# ADAPTIVE CAPACITY ALLOCATION IN MPLS NETWORKS

by

**Hesham S. Bin-Abbas**

B.S. in E.E., King Saud University, Riyadh, 1993

M.S. in E.E., University of Washington, Seattle, 1996

Submited to the Graduate Faculty of

the School of Engineering in partial fulfillment

of the requirements for the degree of

**Doctor of Philosophy**

University of Pittsburgh

2004

UNIVERSITY OF PITTSBURGH

SCHOOL OF ENGINEERING

This dissertation was presented

by

Hesham S. Bin-Abbas

It was defended on

July 22, 2004

and approved by

David Tipper, Ph.D., Associate Professor

Marlin Mickle, Ph.D., Professor

Luis Chaparro, Ph.D., Associate Professor

Ching-Chung Li, Ph.D., Professor

Prashant Krishnamurthy, Ph.D., Assistant Professor

Dissertation Advisors: David Tipper, Ph.D., Associate Professor,

Amro El-Jaroudi, Ph.D., Associate Professor

**ADAPTIVE CAPACITY ALLOCATION IN MPLS NETWORKS**

Hesham S. Bin-Abbas, PhD

University of Pittsburgh, 2004

Traffic Congestion is one of the salient issues that affects overall network performance. Network traffic has become very dynamic due to a variety of factors, such as, the number of users varies with time of the day, multimedia applications, bursts in traffic due to a failure and so on. Recently, Multi-Protocol Label Switching (MPLS) networks have emerged as a technology with many promising features such as traffic engineering, QoS provisioning, and speeding up the traffic transmission. However, MPLS still suffers from the nonstationary/transient conditions that sometimes cause congestion. Actually, congestion does not always occur when the network is short capacity, but rather, when the network resources are not efficiently utilized. Thus, it is very important to develop an algorithm that efficiently and dynamically adjusts the available capacity.

In this thesis, we propose an adaptive capacity allocation scheme. We have started our consideration with a single traffic class system that has dynamic traffic where traffic arrival is considered at the level of connection/call arrival. We assume that the virtual network for this traffic class operates as a loss system; i.e. if a connection does not find bandwidth, the connection is blocked and cleared from the system. Then, we extended our work to include the multiple traffic classes. Two cases have been studied and analyzed; when classes have no coupling and when they are coupled.

The capacity allocation scheme is derived from a first-order, differential equation-based, fluid-flow model that captures the traffic dynamics. The scheme aims to maintain the connection blocking probability within a specified range by dynamically adjusting the allocated capacity. A fluid flow differential equation model is developed to model the changing traffic environment. Using the fluid flow model, Lyapunov Stability theory is used to derive a novel adaptive capacity adjustment scheme which guarantees overall system stability while maintaining the target QoS parameters. Numerical results are given which show that the Lyapunov control based scheme successfully provides the desired QoS requirements and performs better than existing schemes in the literature.

## DESCRIPTORS

Adaptive Bandwidth Allocation
Capacity Control
Dynamic Traffic
Fluid Flow Differential Equation
Lyapunov Stability Theorem
Multi Protocol Label Switching
PSFFA
Quality of Service

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

## PREFACE

First and foremost, I would like to thank my research advisor, Dr. David Tipper, and my academic advisor, Dr. Amro El-Jaroudi. I am grateful to Dr. Tipper for his support and effort in providing invaluable guidance and inspired ideas throughout the preparation of this research. I must also thank Dr. Amro El-jaroudi for his cooperation and facilitating my educational mission. I sincerely appreciate all of their advice, support, and assistance, without which this thesis would not have been possible.

I also would like to thank Dr. Luis Chaparro, Dr. Marlin Mickle, Dr. C-C Li, and Dr. Prashant Krishnamurthy, for serving on my committee and providing me with their thoughtful input and comments.

I am grateful to my family that joined me in traveling overseas to support me in my educational pursuits. Special thanks go to my lovely wife for her patience and encouragement during the bad times, and to my two children: Dania, who suffered so much loneliness here and isolation away from her relatives, and to my young son, whom I was too busy to give the attention he deserved. As their feelings of homesickness increased, it doubled the pressure that I felt, but it also served as an essential motivation to finish my dissertation.

I'd like to extend my thanks to my family back home: my two brothers, Eng. Esam and Dr. Bassam, and my sister, Dr. Elham for their encouragement and readiness to help. Most importantly I have to thank my mother, to whom I am indebted for her care, love and endless

prayers for me. Special thanks go out to the person who taught me more than anyone else, my father, who died just before starting my Ph.D.s.

Never have I wished that he were alive more than right now so that I could see the joy in his eyes.  May God bless him with his mercy.

Last and not least, I want to acknowledge the financial support from King Abdulaziz City for Science and Technology, Saudi Arabia that offered me this opportunity to complete my study.

# 1.0 INTRODUCTION

## 1.1 PRELIMINARIES

One of the most pressing challenges in designing IP networks is the provisioning of Quality of Service (QoS). The current Internet operates in a best-effort manner, which is considered insufficient for applications demanding QoS. In addition, as Internet migrates to commercial enterprise, providing reliable QoS may well become a crucial factor in influencing the customer's penchant to pay for network services. Consequently, Internet Engineering Task Force (IETF) working groups have proposed several frameworks and mechanisms for QoS support such as Integrated Services (IntServ) and RSVP framework, and Differentiated Services (DiffServ) framework. The latter is more dominant due to its characteristics of scalability. These two models provide a framework for service classification, differentiation and prioritization, with the objective to guarantee network performance and satisfy customers' requirements. Also, IETF has proposed a relatively new technology known as Multi-Protocol Label Switching (MPLS) framework. MPLS provides flexible and high speed packet transfer capabilities by defining a new header, which is composed of a stack of fixed-length labels, and inserted between layer 2 and layer 3 headers. A contiguous set of nodes on which packets are transferred by labels swapping constitutes a data forwarding path referred to as a Label Switching Path (LSP). Although MPLS was originally designed as a way of improving the forwarding speed of routers, it is now emerging as a crucial standard technology that offers new capabilities of providing the objective of QoS for large-scale IP network. DiffServ together with MPLS provide a powerful and highly scalable framework for QoS provisioning in IP networks; MPLS controls the data path of each traffic class while DiffServ controls the QoS differentiation in each node.

While these frameworks, especially MPLS and DiffServ, have shown great promise in achieving some performance objectives in MPLS networks such as high service quality, efficiency, survivability, but other critical issues remain such as the congestion control. Congestion does not always occur when the network is lacking of network capacity, but rather, when the network resources are not efficiently utilized. Typical cases causing network congestion, for example, are when the network is designed with assumption of steady-state parameters, or inefficient bandwidth sharing mechanism is deployed. Therefore, an effective bandwidth allocation scheme that works under both transient and steady-state conditions is needed.

## 1.2 PROBLEM STATEMENT

Network traffic tends to be dynamic due to a variety of factors, such as, load sharing, changes in routing and flow control parameters, failure of links, nodes or other network resources and most commonly, nonstationary input loads. An important network management issue is how to maintain QoS requirements in this dynamic environment. One simple way to provide QoS is by allocating capacity for the highest load over a time window. However, the downside of this approach is that the capacity may be vastly under utilized when the load is significantly below the peak value within that time window. Although several researches have dealt with the problem of capacity allocation, the majority of the work considered the systems under only the steady state conditions, which make them inaccurate for nonstationary/transient cases. Therefore, the problem that we are trying to solve is to develop an adaptive scheme that can dynamically allocate and deallocate capacity to maintain QoS requirement for different traffic classes in the nonstationary environment.

## 1.3 THESIS OUTLINE

The main idea of building such dynamic scheme is to take the advantage of the concept of virtual networks that are dynamically reconfigurable. In this context, the capacity can be allocated and deallocated from the underlying transport network for different virtual LSPs to respond to network traffic changes so as to meet, and continue to meet, QoS requirements. In other words, if the traffic is low for a particular service class, at a certain point of time, then, as long as QoS is met, some capacity may be deallocated for use by other services or assigned to a shared pool for use by other services. Similarly, the reverse situation is also possible, i.e., if network resources are affected due to a component failure in the virtual network for a certain traffic class, then additional capacity to maintain acceptable QoS under the failure situation may need to be requested from the transport network. Such dynamic capacity adjustment, in response to or in anticipation of changes in traffic, can lead to better use of network resources, since idle capacity for a service class can be deallocated for use by another service class.

In this thesis, we take a control theoretic approach to solving the adaptive capacity allocation problem. The proposed scheme is derived from a first-order, differential equation-based, fluid-flow model that captures the traffic dynamics. First, we have considered a single traffic class system that has dynamic traffic where traffic arrival is considered at the level of connection/call arrival. Thus, the QoS parameter of interest is the connection/call blocking of the service class. We assume that a connection is admitted to the system only if the network can provide within the connection QoS guarantee along with the bandwidth requirement. More specifically, we have designed an algorithm that computes the blocking probability in every iteration under the dynamic environment and accordingly adjust the capacity to keep the blocking rate within range to meet the QoS service requirements. Using the fluid flow model, Lyapunov Stability theory is used to derive a novel adaptive capacity adjustment scheme which guarantees overall system stability while maintaining the target QoS parameters.

Secondly, the work is extended to implement a scheme for adaptive capacity allocation for multiple service classes. Two cases have been considered; when the traffics have no couplings, in which each traffic class has its dedicated share of bandwidth, and when they are coupled in which the link bandwidth is completely shared and available for all incoming traffics. Although there are two traffic classes associated with service differentiation in MPLS networks, we provide an algorithm for three traffic classes for the sake of completeness.

We conclude our work with numerical results which show that the Lyapunov control based scheme successfully provides the desired QoS requirements and performs better than existing schemes in the literature.

## 1.4 CONTRIBUTION

The main contribution of this work is the development of a scheme that adaptively adjusts the available link bandwidth according to the offered load instead of the current technique of the static resource allocation, in which the capacity is allocated based on the peak offered load. The significance of the developed scheme is illustrated when compared with the traditional connection admission control as shown in chapter 4. With the adaptive capacity allocation, new arrivals can be admitted to the system when connections have low utilization for the same time window while maintaining the required QoS.

Although the MPLS network is considered as an application, the obtained scheme is generic and can be applied to different platforms where dynamic bandwidth sharing is appropriate such as cellular wireless networks.

# 2.0 BACKGROUND AND LITERATURE SURVEY

## 2.1 LITERATURE REVIEW

### 2.1.1 Introduction

Resource management and efficient Bandwidth Allocation have been receiving a great attention for the past few years since the need of excess capacity is not always desirable or even necessary. With efficient link sharing schemes, one can avoid congestions, increase the overall utilization, and provide the network services with the required QoS. Therefore, several researches have focused on this problem from different perspectives; some models dealt with problem of the capacity allocation at the packet level, whilst others consider it at the connection level. The next section briefly provides a review of the different schemes of capacity allocation at the packet level. In the following section, other schemes that address the same problem but at the connection level are presented and discussed. The chapter concludes with the problem statement of this thesis.

### 2.1.2 Capacity Allocation at the Packet Level

Several models of Adaptive Bandwidth Control (ABC) at the packet-level for a single queue have been proposed and are based on the model depicted in Figure (1).

Existing ABC algorithms can be classified according to the underlying control technique used, or according to the QoS metrics guaranteed, as shown in Figure(2).

Regarding the underlying control technique, ABC schemes can roughly be classified as being either closed-loop or open-loop. The closed-loop, or feedback control, approach arises

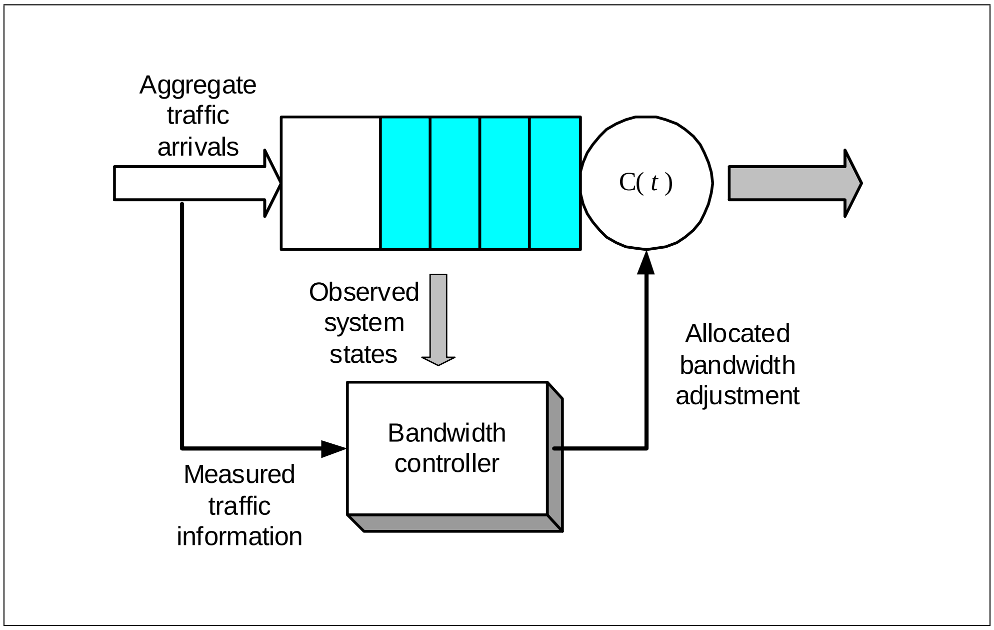


Figure 1: A model of adaptive bandwidth control for a single queue.

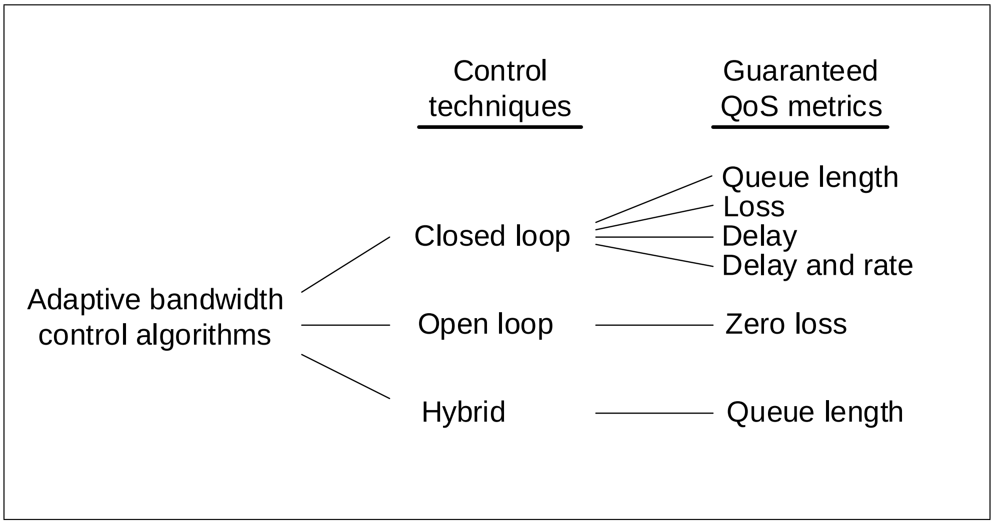


Figure 2: Classification of ABC algorithms that guarantee packet-level QoS metrics.

naturally in this context where the packet loss, average queue length, or other system states, are regularly observed to provide the feedback to adjust the allocated bandwidth. The closed-loop control approach can be further categorized based on the guaranteed QoS metrics, including the average queue length [5], [6], loss [7]–[10], and delay [11], [12]. The open-loop control approach involves predicting the input traffic rate using the past history. The service rate is then adjusted to match the predicted rate to attain zero packet loss or low queuing delay. However, achieving a given target QoS using open-loop control is difficult due to the lack of explicit relationship between the predicted traffic rate and the target QoS. Consequently, most of the existing work for open-loop ABC only attempts to deliver very low or zero packet loss rather than guarantee it quantitatively [13]–[15]. The hybrid of feedback control and open-loop control is also possible to eliminate the drawbacks found in both approaches [16]. More discussions and evaluations are available in [17].

### 2.1.3 Capacity Allocation at the Connection Level

For the call admission control, various bandwidth access schemes such as complete sharing, complete partitioning, partial sharing, and classical trunk reservation schemes have been proposed and studied in the literature for multirate circuit switched networks and circuit-mode traffic [18]-[31]. However, most of work in those approaches is to provide analytic models to consider the impact of control schemes on connection blocking of individual traffic streams. Unfortunately, these schemes either do not consider the concept of capacity adjustments techniques or they assume stationary traffic conditions.

On the other hand, analyzing network performance under nonstationary traffic has been addressed for a queuing system using a fluid- flow-based approach [33], [48], [49]. A good summary of different ideas can be found in [49]. In Qian's, et. al. [46] work, admission control with non-stationary traffic is addressed using Chapman-Kolmogorov's equation for a multi-rate loss system. However, none of these work considers the dynamic adjustment of available capacity. There have been some work that addresses connection blocking and capacity adjustment based on changing traffic conditions [32], [36], [41], [47]. From a network dimensioning perspective, it has been shown in [42] that the overall network capacity requirement is less under a dynamic traffic condition using the dynamic virtual path concept in ATM networks as compared to static allocation. In [37], a dynamically capacity environment is addressed, and several capacity adjustment control schemes have been presented. These schemes make use of the Fluid-Flow Approximation for analyzing a dynamic traffic in a single traffic class loss system based on the blocking and utilization as means to calculate when and how much adjustment should be made. However, the drawback of those schemes is the assumption of a priori knowledge about the traffic, for which the parameters of adjustment can be properly obtained and supplied. Therefore, it is vulnerable to failure in some cases.

## 2.2 MPLS OVERVIEW

Over the last few years, the Internet has evolved into a ubiquitous network and inspired the development of a variety of new applications in business and consumer markets. These new applications have driven the demand for increased and guaranteed bandwidth requirements in the backbone of the network. In addition to the traditional data services currently provided over the Internet, new voice and multimedia services are being developed and deployed. The Internet has emerged as the network of choice for providing these converged services. However, the demands placed on the network by these new applications and services, in terms of speed and bandwidth, have strained the resources of the existing Internet infrastructure. This transformation of the telecommunication network infrastructure toward a packet- and cell-based infrastructure has introduced uncertainty into what has traditionally been a fairly deterministic network.

In addition to the issue of resource constraints, another challenge relates to the transport of traffic over the backbone to provide differentiated classes of service to users. The exponential growth in the number of users and the volume of traffic adds another dimension to this problem. Class of service (CoS) and QoS issues must be addressed to in order to efficiently support the diverse requirements of the wide range of network users.

Multi-protocol label switching (MPLS) has emerged as a versatile solution to address the problems faced by present-day networks—speed, scalability, quality-of-service (QoS) provisioning, bandwidth-management, and traffic engineering. It is believed that MPLS is going to play an important role in the routing, switching, and forwarding of packets through the next-generation network in order to meet the service demands of the network users.

This chapter reviewes the basic architecture of the MPLS network as described in [1], and more information can be found in [2]-[4].

### 2.2.1 Traditional Routing and Packet Switching

The initial deployment of the Internet addressed the requirements of data transfer over the network. This network catered to simple applications such as file transfer and remote login. To carry out these requirements, a simple software-based router platform, with network interfaces to support the existing T1/E1– or T3/E3–based backbones, was sufficient. As the demand for higher speed and the ability to support higher-bandwidth transmission rates emerged, devices with capabilities to switch at the Level-2 (data link) and the Level-3 (network layer) in hardware had to be deployed. Layer-2 switching devices addressed the switching bottlenecks within the subnets of a local-area network (LAN) environment. Layer-3 switching devices helped alleviate the bottleneck in Layer-3 routing by moving the route lookup for Layer-3 forwarding to high-speed switching hardware.

These early solutions addressed the need for wire-speed transfer of packets as they traversed the network, but they did not address the service requirements of the information contained in the packets. Also, most of the routing protocols deployed today are based on algorithms designed to obtain the shortest path in the network for packet traversal and do not take into account additional metrics (such as delay, jitter, and traffic congestion), which can further degrade the network performance. Traffic engineering is a challenge for network managers.

### 2.2.2 MPLS Components

MPLS is an Internet Engineering Task Force (IETF)–specified framework that provides for the efficient designation, routing, forwarding, and switching of traffic flows through the network.

MPLS performs the following functions:

- specifies mechanisms to manage traffic flows of various granularities, such as flows between different hardware, machines, or even flows between different applications.
- remains independent of the Layer-2 and Layer-3 protocols.
- provides a means to map IP addresses to simple, fixed-length labels used by different packet-forwarding and packet-switching technologies.
- interfaces to existing routing protocols such as resource reservation protocol (RSVP) and open shortest path first (OSPF).
- supports the IP, ATM, and frame-relay Layer-2 protocols.

In MPLS, data transmission occurs on label-switched paths (LSPs). LSPs are a sequence of labels at each and every node along the path from the source to the destination. LSPs are established either prior to data transmission (control-driven) or upon detection of a certain flow of data (data-driven). The labels, which are underlying protocol-specific identifiers, are distributed using either label distribution protocol (LDP) or RSVP or piggybacked on routing protocols like border gateway protocol (BGP) and OSPF. Each data packet encapsulates and carries the labels during their journey from source to destination. High-speed switching of data is possible because the fixed-length labels are inserted at the very beginning of the packet or cell and can be used by hardware to switch packets quickly between links.

**2.2.2.1 Label Switching Routers and Label Edge Routers** The devices that participate in the MPLS protocol mechanisms can be classified into label edge routers (LERs) and label switching routers (LSRs).

An LSR is a high-speed router device in the core of an MPLS network that participates in the establishment of LSPs using the appropriate label signaling protocol and high-speed switching of the data traffic based on the established paths.

An LER is a device that operates at the edge of the access network and MPLS network. LERs support multiple ports connected to dissimilar networks (such as frame relay, ATM, and Ethernet) and forwards this traffic on to the MPLS network after establishing LSPs, using the label signaling protocol at the ingress and distributing the traffic back to the access networks at the egress. The LER plays a very important role in the assignment and removal of labels, as traffic enters or exits an MPLS network.

**2.2.2.2 Forward Equivalent Class** The forward equivalence class (FEC) is a representation of a group of packets that share the same requirements for their transport. All packets in such a group are provided the same treatment en route to the destination. As opposed to conventional IP forwarding, in MPLS, the assignment of a particular packet to a particular FEC is done just once, as the packet enters the network. FECs are based on service requirements for a given set of packets or simply for an address prefix. Each LSR builds a table to specify how a packet must be forwarded. This table, called a label information base (LIB), is comprised of FEC–to-label bindings.

**2.2.2.3 Labels and Label Bindings** A label, in its simplest form, identifies the path a packet should traverse. A label is carried or encapsulated in a Layer-2 header along with the packet. The receiving router examines the packet for its label content to determine the next hop. Once a packet has been labeled, the rest of the journey of the packet through the backbone is based on label switching. The label values are of local significance only, meaning that they pertain only to hops between LSRs.

Once a packet has been classified as a new or existing FEC, a label is assigned to the packet. The label values are derived from the underlying data link layer. For data link layers (such as frame relay or ATM), Layer-2 identifiers, such as data link connection identifiers (DLCIs) in the case of frame-relay networks or virtual path identifiers (VPIs)/virtual channel identifiers (VCIs) in case of ATM networks, can be used directly as labels. The packets are then forwarded based on their label value.

Labels are bound to an FEC as a result of some event or policy that indicates a need for such binding. These events can be either data-driven bindings or control-driven bindings. The latter is preferable because of its advanced scaling properties that can be used in MPLS.

Label assignment decisions may be based on forwarding criteria such as the following:

- Destination uni-cast routing.
- Traffic engineering.

- Multicast.
- Virtual private network (VPN).
- QoS.

The generic label format is illustrated in Figure (3). The label is contained in a special header called "shim header". This header consists of 32 bits in four parts – twenty bits are used for the label, three bits for experimental functions, one bit for stack function, and eight bits for time to live (TTL). The label can be either embedded in the header of the data link layer (the ATM VCI/VPI shown in Figure (4) and the frame-relay DLCI shown in Figure (5), or in the shim (between the Layer-2 data-link header and Layer-3 network layer header, as shown in Figure (6).

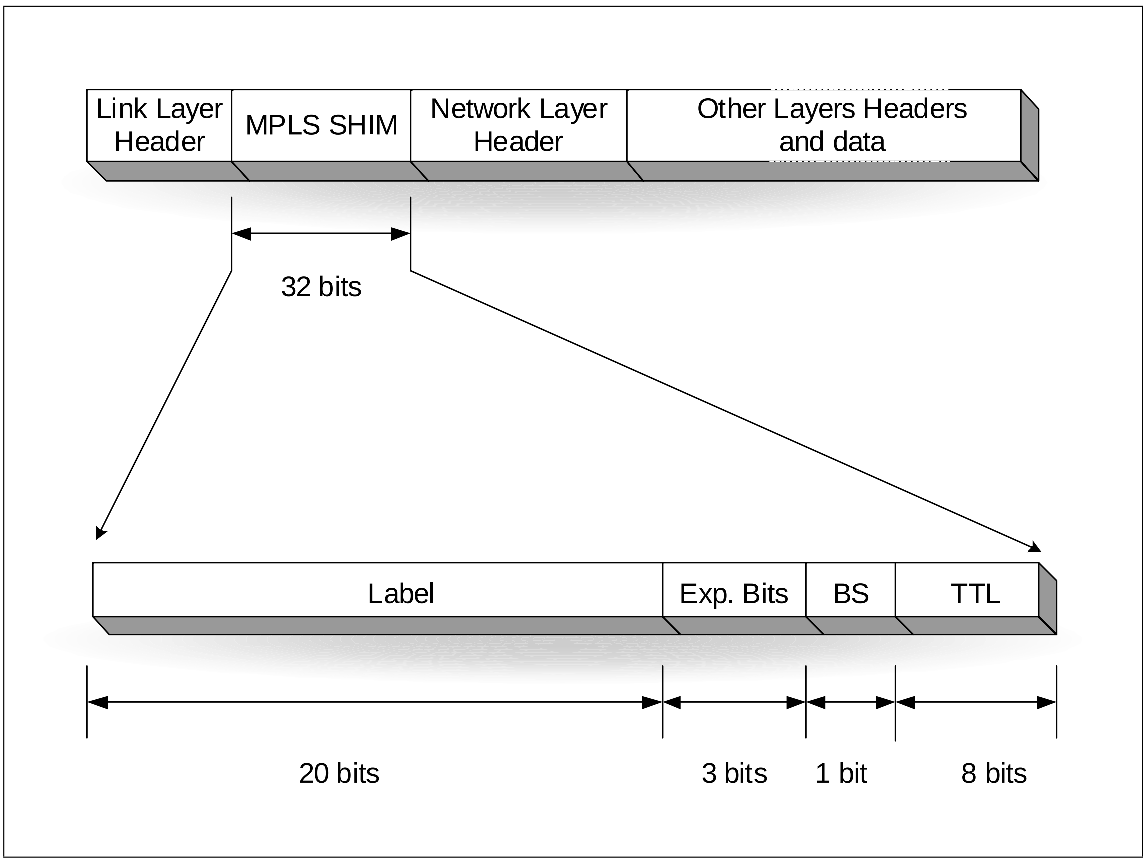


Figure 3: MPLS Generic Label Format

#### 2.2.2.4 Label Creation

There are several methods used in label creation:

- Topology-based method—uses normal processing of routing protocols (such as OSPF and BGP).
- Request-based method—uses processing of request-based control traffic (such as RSVP).
- Traffic-based method—uses the reception of a packet to trigger the assignment and distribution of a label.

The topology- and request-based methods are examples of control-driven label bindings, while the traffic-based method is an example of data-driven bindings.

#### 2.2.2.5 Label Distribution

The MPLS architecture does not mandate a single method of signaling for label distribution. Existing routing protocols, such as the Border Gateway Protocol (BGP), have been enhanced to piggyback the label information within the contents of the protocol. The RSVP has also been extended to support piggybacked exchange of labels. The Internet Engineering Task Force (IETF) has also defined a new protocol known as the label distribution protocol (LDP) for explicit signaling and management of the label space. Extensions to the base LDP protocol have also been defined to support explicit routing based on QoS and CoS requirements. These extensions are captured in the constraint-based routing (CR)–LDP protocol definition.

A summary of the various schemes for label exchange is as follows:

- LDP—maps unicast IP destinations into labels
- RSVP, CR–LDP—used for traffic engineering and resource reservation
- Protocol-independent multicast (PIM)—used for multicast states label mapping
- BGP—external labels (VPN)

#### 2.2.2.6 Label-Switched Paths (LSPs)

**A collection of MPLS**—enabled devices represents an MPLS domain. Within an MPLS domain, a path is set up for a given packet to travel based on an FEC. The LSP is set up prior to data transmission. MPLS provides the following two options to set up an LSP.

- **Hop-by-hop routing**—Each LSR independently selects the next hop for a given FEC. This methodology is similar to that currently used in IP networks. The LSR uses any available routing protocols, such as OSPF, ATM private network-to-network interface (PNNI), etc.
- **Explicit routing (ER)**—Explicit routing is similar to source routing. The ingress LSR (i.e., the LSR where the data flow to the network first starts) specifies the list of nodes through which the ER–LSP traverses. The path specified could be non-optimal, as well. Along the path, the resources may be reserved to ensure QoS to the data traffic. This eases traffic engineering throughout the network, and differentiated services can be provided using flows based on policies or network management methods.

The LSP setup for an FEC is unidirectional in nature. The return traffic must take another LSP.

**2.2.2.7 Label Spaces** The labels used by an LSR for FEC–label bindings are categorized as follows:

- **Per platform**—the label values are unique across the whole LSR. The labels are allocated from a common pool. No two labels distributed on different interfaces have the same value.
- **Per interface**—the label ranges are associated with interfaces. Multiple label pools are defined for interfaces, and the labels provided on those interfaces are allocated from the separate pools. The label values provided on different interfaces could be the same.

**2.2.2.8 Label Merging** The incoming streams of traffic from different interfaces can be merged together and switched using a common label if they are traversing the network toward the same final destination. This is known as stream merging or aggregation of flows.

If the underlying transport network is an ATM network, LSRs could employ virtual path (VP) or virtual channel (VC) merging. In this scenario, cell interleaving problems, which arise when multiple streams of traffic are merged in the ATM network, need to be avoided.

**2.2.2.9 Label Retention** MPLS defines the treatment for label bindings received from LSRs that are not the next hop for a given FEC. Two modes are defined.

- **Conservative**—in this mode, the bindings between a label and an FEC received from LSRs that are not the next hop for a given FEC are discarded. This mode requires an LSR to maintain fewer labels. This is the recommended mode for ATM–LSRs.
- **Liberal**—in this mode, the bindings between a label and an FEC received from LSRs that are not the next hop for a given FEC are retained. This mode allows for quicker adaptation to topology changes and allows for the switching of traffic to other LSPs in case of changes.

**2.2.2.10 Label Control** MPLS defines modes for distribution of labels to neighboring LSRs.

- **Independent**—in this mode, an LSR recognizes a particular FEC and makes the decision to bind a label to the FEC independently to distribute the binding to its peers. The new FECs are recognized whenever new routes become visible to the router.
- **Ordered**—in this mode, an LSR binds a label to a particular FEC if and only if it is the egress router or it has received a label binding for the FEC from its next hop LSR. This mode is recommended for ATM–LSRs.

### 2.2.3 Signaling Mechanisms

- **Label request**—using this mechanism, an LSR requests a label from its downstream neighbor so that it can bind to a specific FEC. This mechanism can be employed down the chain of LSRs up until the egress LER (i.e., the point at which the packet exits the MPLS domain).
- **Label mapping**—in response to a label request, a downstream LSR will send a label to the upstream initiator using the label mapping mechanism.

The above concepts for label request and label mapping are explained in an example shown in Figure (7).

### 2.2.4 Label Distribution Protocol (LDP)

The LDP is a new protocol for the distribution of label binding information to LSRs in an MPLS network. It is used to map FECs to labels, which, in turn, create LSPs. LDP sessions are established between LDP peers in the MPLS network (not necessarily adjacent). The peers exchange the following types of LDP messages:

- **discovery messages**—announce and maintain the presence of an LSR in a network
- **session messages**—establish, maintain, and terminate sessions between LDP peers
- **advertisement messages**—create, change, and delete label mappings for FECs
- **notification messages**—provide advisory information and signal error information

### 2.2.5 Label Stack

The label stack mechanism allows for hierarchical operation in the MPLS domain. It basically allows MPLS to be used simultaneously for routing at the fine-grain level (e.g., between individual routers within an Internet service provider [ISP] and at a higher domain-by-domain level). Each level in a label stack pertains to some hierarchical level. This facilitates a tunneling mode of operation in MPLS.

### 2.2.6 Traffic Engineering

Traffic engineering is a process that enhances overall network utilization by attempting to create a uniform or differentiated distribution of traffic throughout the network. An important result of this process is the avoidance of congestion on any one path. It is important to note that traffic engineering does not necessarily select the shortest path between two devices. It is possible that, for two packet data flows, the packets may traverse completely different paths even though their originating node and the final destination node are the same. This way, the less-exposed or less-used network segments can be used and differentiated services can be provided.

In MPLS, traffic engineering is inherently provided using explicitly routed paths. The LSPs are created independently, specifying different paths that are based on user-defined policies. However, this may require extensive operator intervention. RSVP and CR–LDP are two possible approaches to supply dynamic traffic engineering and QoS in MPLS.

### 2.2.7 Constraint-based Routing

Constraint-based routing (CR) takes into account parameters, such as link characteristics (bandwidth, delay, etc.), hop count, and QoS. The LSPs that are established could be CR–LSPs, where the constraints could be explicit hops or QoS requirements. Explicit hops dictate which path is to be taken. QoS requirements dictate which links and queuing or scheduling mechanisms are to be employed for the flow.

When using CR, it is entirely possible that a longer (in terms of cost) but less loaded path is selected. However, while CR increases network utilization, it adds more complexity to routing calculations, as the path selected must satisfy the QoS requirements of the LSP. CR can be used in conjunction with MPLS to set up LSPs. The IETF has defined a CR–LDP component to facilitate constraint-based routes.

### 2.2.8 MPLS Operation

The following steps must be taken for a data packet to travel through an MPLS domain.

- label creation and distribution
- table creation at each router
- label-switched path creation
- label insertion/table lookup
- packet forwarding

The source sends its data to the destination. In an MPLS domain, not all of the source traffic is necessarily transported through the same path. Depending on the traffic characteristics, different LSPs could be created for packets with different CoS requirements.

In Figure (8), LER1 is the ingress and LER4 is the egress router. Moreover, Table 1 illustrates the step-by-step MPLS operations that occur on the data packets in an MPLS domain. Table 2 shows a simple example of the LIB tables.

It is interesting to consider the example of two streams of data packets entering an MPLS domain:

- One packet stream is a regular data exchange between servers (e.g., file transfer protocol [FTP]).
- The other packet stream is an intensive video stream, which requires the traffic engineering parameters of QoS (e.g., videoconferencing ).
- These packet streams are classified into 2 separate FECs at the ingress LSR.
- The label mappings associated with the streams are 3 and 9, respectively.
- The input ports at the LSR are 1 and 2, respectively.
- The corresponding output interfaces are 3 and 1, respectively.
- Label swapping must also be done, and the previous labels must be exchanged for 6 and 7, respectively.

### 2.2.9 MPLS Protocol Stack Architecture

The core MPLS components can be broken down into the following parts:

- Network layer (IP) routing protocols
- Edge of network layer forwarding
- Core network label-based switching
- Label schematics and granularity
- Signaling protocol for label distribution
- Traffic engineering
- Compatibility with various Layer-2 forwarding paradigms (ATM, frame relay, PPP)

Figure (9) depicts the protocols that can be used for MPLS operations. The routing module can be any one of several popular industry protocols. Depending on the operating environment, the routing module can be OSPF, BGP, or ATM's PNNI, etc. The LDP module utilizes transmission control protocol (TCP) for reliable transmission of control data from one LSR to another during a session. The LDP also maintains the LIB. The LDP uses the user datagram protocol (UDP) during its discovery phase of operation. In this phase, the

LSR tries to identify neighboring elements and also signals its own presence to the network. This is done through an exchange of hello packets.

The IP Fwd is the classic IP–forwarding module that looks up the next hop by matching the longest address in its tables. For MPLS, this is done by LERs only. The MPLS Fwd is the MPLS forwarding module that matches a label to an outgoing port for a given packet. The layers, shown in the box with the broken line, can be implemented in hardware for fast, efficient operation.

### 2.2.10 MPLS Applications

MPLS addresses today's network backbone requirements effectively by providing a standards-based solution that accomplishes the following:

- Improves packet-forwarding performance in the network
  - MPLS enhances and simplifies packet forwarding through routers using Layer-2 switching paradigms.
  - MPLS is simple, which allows for easy implementation.
  - MPLS increases network performance because it enables routing by switching at wireline speeds.
- Supports QoS and CoS for service differentiation
  - MPLS uses traffic-engineered path setup and helps achieve service-level guarantees.
  - MPLS incorporates provisions for constraint-based and explicit path setup.

- Supports network scalability
  - MPLS can be used to avoid the N2 overlay problem associated with meshed IP–ATM networks.
- Integrates IP and ATM in the network
  - MPLS provides a bridge between access IP and core ATM.

  - MPLS can reuse existing router/ATM switch hardware, effectively joining the two disparate networks.

- Builds interoperable networks
  - MPLS is a standards-based solution that achieves synergy between IP and ATM networks.
  - MPLS facilitates IP–over-synchronous optical network (SONET) integration in optical switching.
  - MPLS helps build scalable VPNs with traffic-engineering capability.

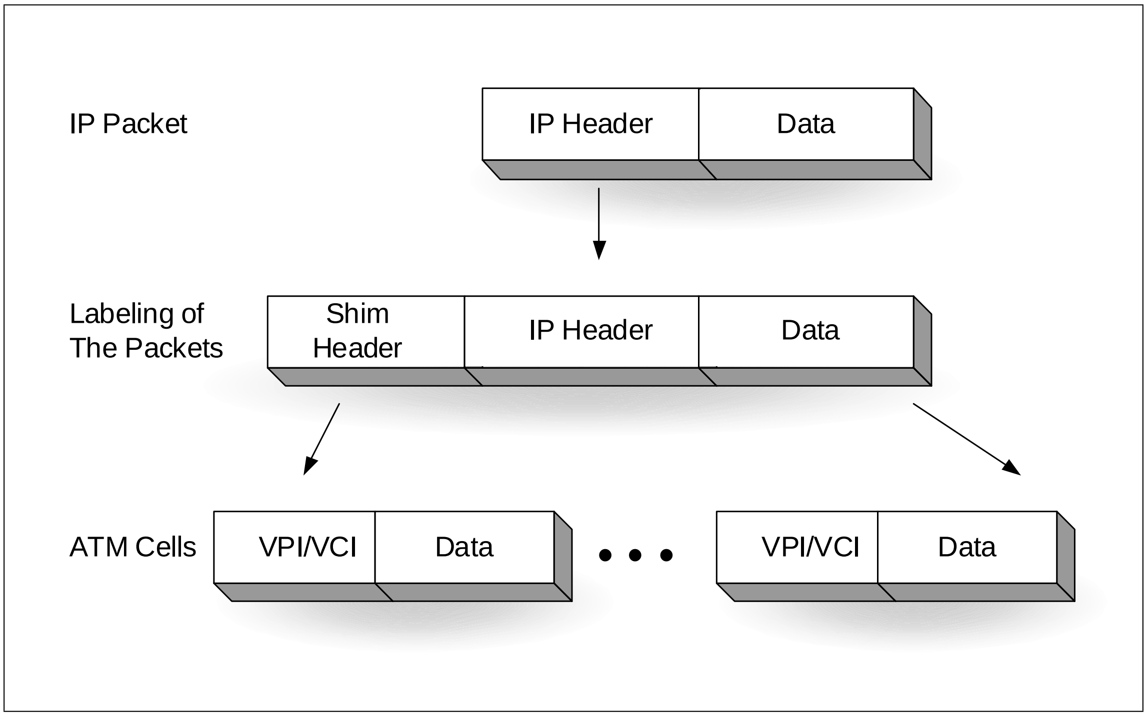


Figure 4: ATM as the Data Link Layer

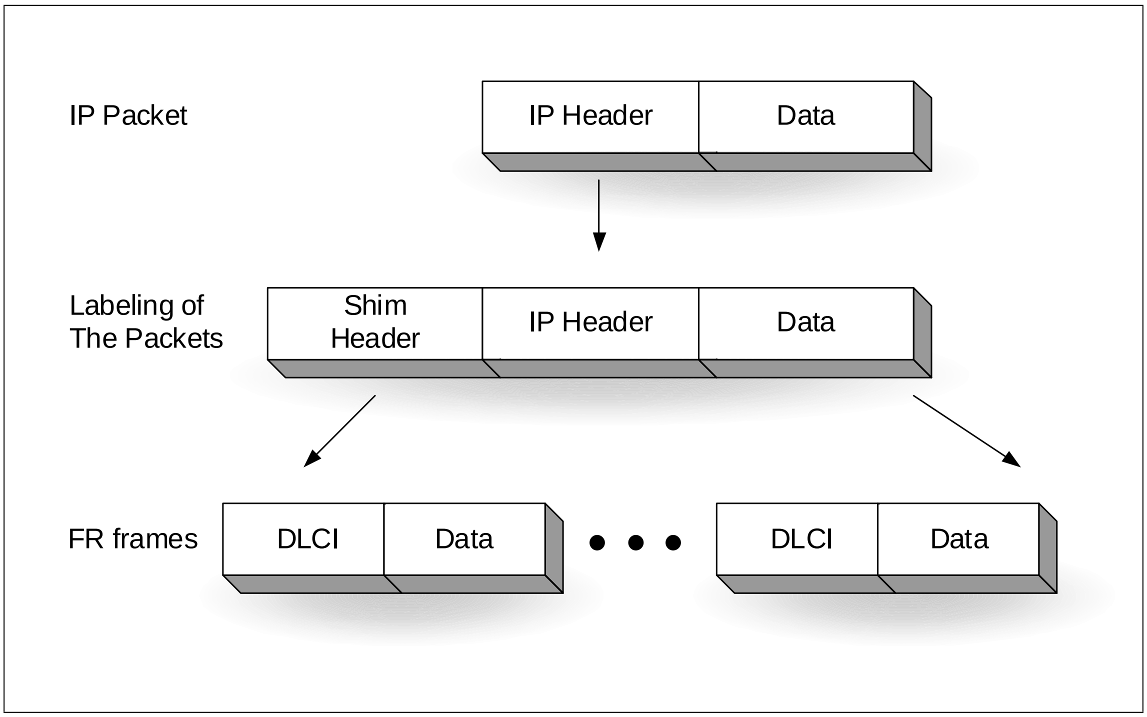


Figure 5: Frame Relay as the Data Link Layer

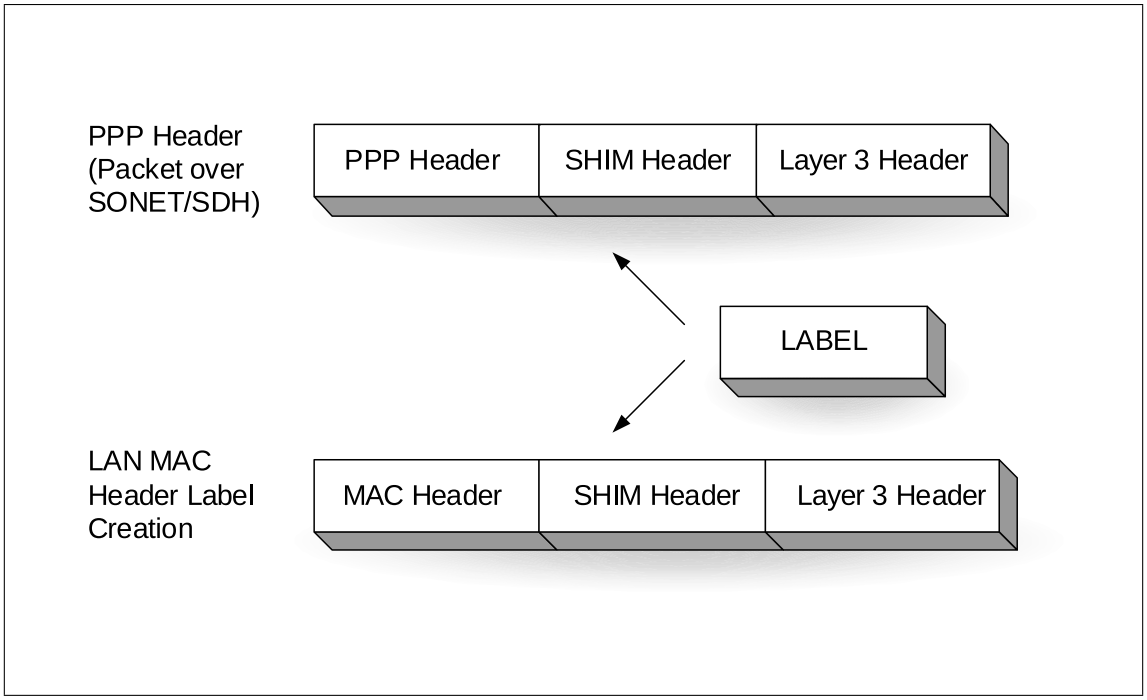


Figure 6: Point-to-Point (PPP)/Ethernet as the data link layer

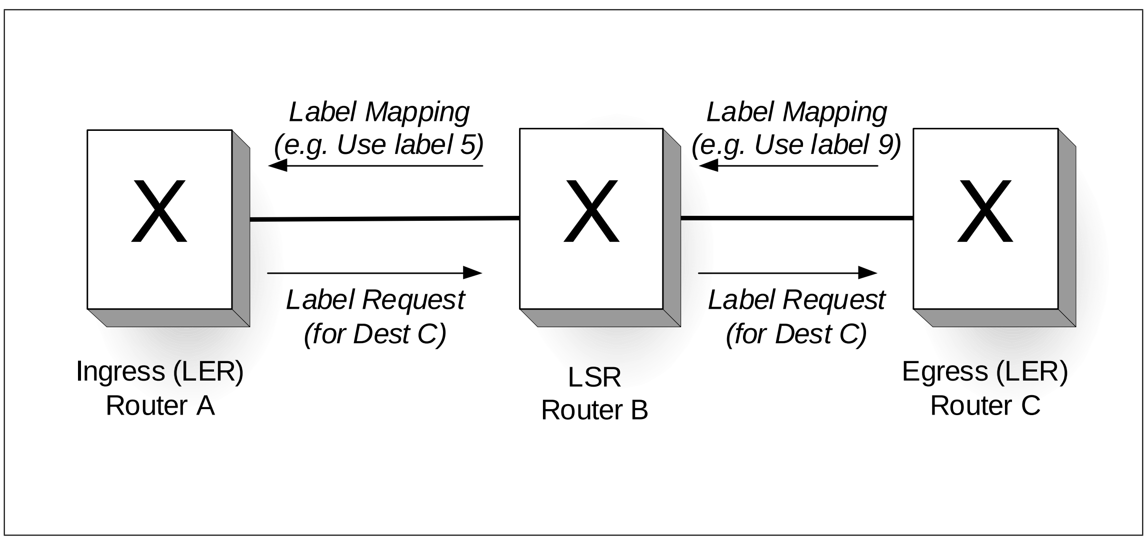


Figure 7: Signalling mechanism

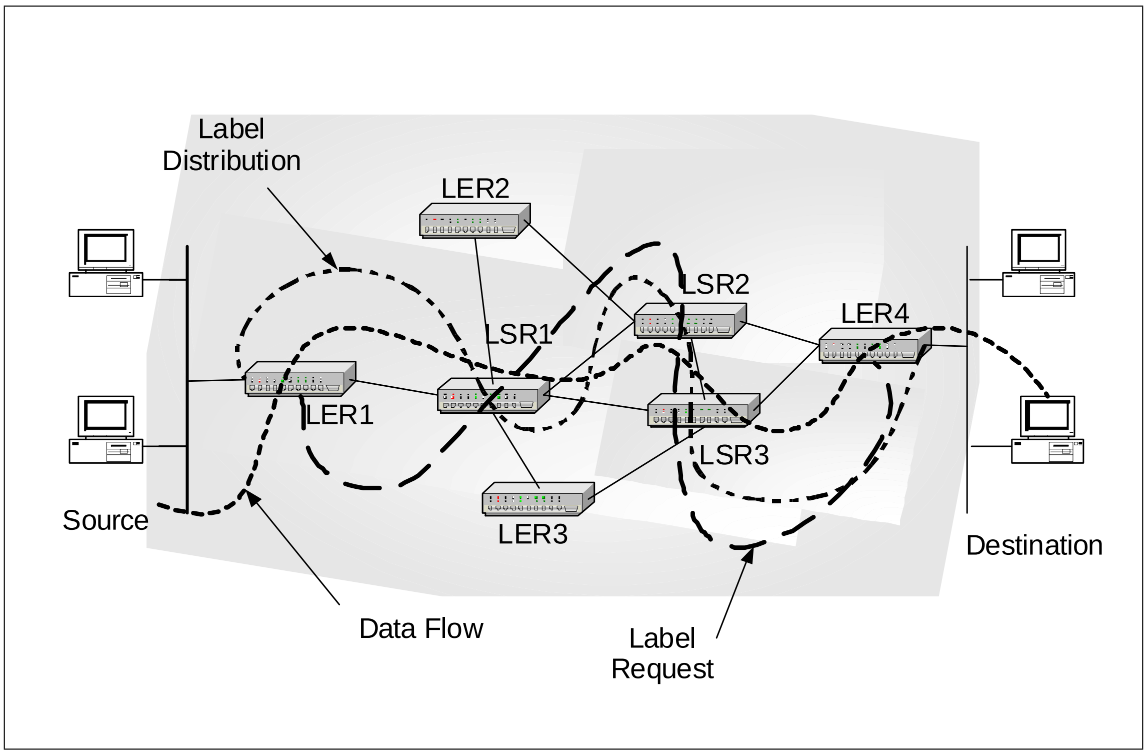


Figure 8: LSP Creation and Packet Forwarding through an MPLS Domain

**Table 1: MPLS Actions**

| MPLS Actions | Description |
|---|---|
| Label creation and Label Distribution | • Before any traffic begins the routers make the decision to bind a label to a specific FEC and build their tables.<br><br>• In LDP, downstream routers initiate the distribution of labels and the label/FEC binding.<br><br>• In addition, traffic-related characteristics and MPLS capabilities are negotiated using LDP.<br><br>• A reliable and ordered transport protocol should be used for the signaling protocol. LDP uses TCP. |
| Table Creation | • On receipt of label bindings each LSR creates entries<br>• in the label information base (LIB).<br><br>• The contents of the table will specify the mapping between a label and an FEC.<br>▪ Mapping between the input port and input label table to the output port and output label table.<br>▪ The entries are updated whenever renegotiation of the label bindings occurs. |
| Label switched path creation | As shown by the dashed lines in Figure (8), the LSPs are created in the reverse direction to the creation of entries in the LIBs. |
| Label insertion/table lookup | • The first router (LER1) in Figure (8) uses the LIB table to find the next hop and request a label for the specific FEC. |

Table 1 (continued)

| | |
|---|---|
| | • Subsequent routers just use the label to find the next hop.<br><br>• Once the packet reaches the egress LSR (LER4), the label is removed and the packet is supplied to the destination. |
| Packet forwarding | With reference to Figure (8), let us examine the path of a packet as it to its destination from LER1, the ingress LSR, to LER4, the egress LSR.<br>1. LER1 may not have any labels for this packet as it is the first occurrence of this request. In an IP network, it will find the longest address match to find the next hop. Let LSR1 be the next hop for LER1.<br>2. LER1 will initiate a label request toward LSR1.<br>3. This request will propagate through the network as indicated by the long broken lines as shown in Figure (8).<br><br>4. Each intermediary router will receive a label from its downstream router starting from LER2 and going upstream till LER1. The LSP setup is indicated by the broken blue lines using LDP or any other signaling protocol. If traffic engineering is required, CR.LDP will be used in determining the actual path setup to ensure the QoS/CoS requirements are complied with<br><br>5. LER1 will insert the label and forward the packet to LSR1.<br><br>6. Each subsequent LSR, i.e., LSR2 and LSR3, will examine the label in the received packet, replace it with the outgoing label and forward it.<br><br>7. When the packet reaches LER4, it will remove the label and deliver it to the destination. |

**Table 1 (continued)**

| | |
|---|---|
| | 8. The actual data path followed by the packet is indicated by the dotted lines. |

**Table 2: Example LIB Table**

| Input Port | Incoming Port Label | Output Port | Outgoing Port Label |
|---|---|---|---|
| 1 | 3 | 3 | 6 |
| 2 | 9 | 1 | 7 |

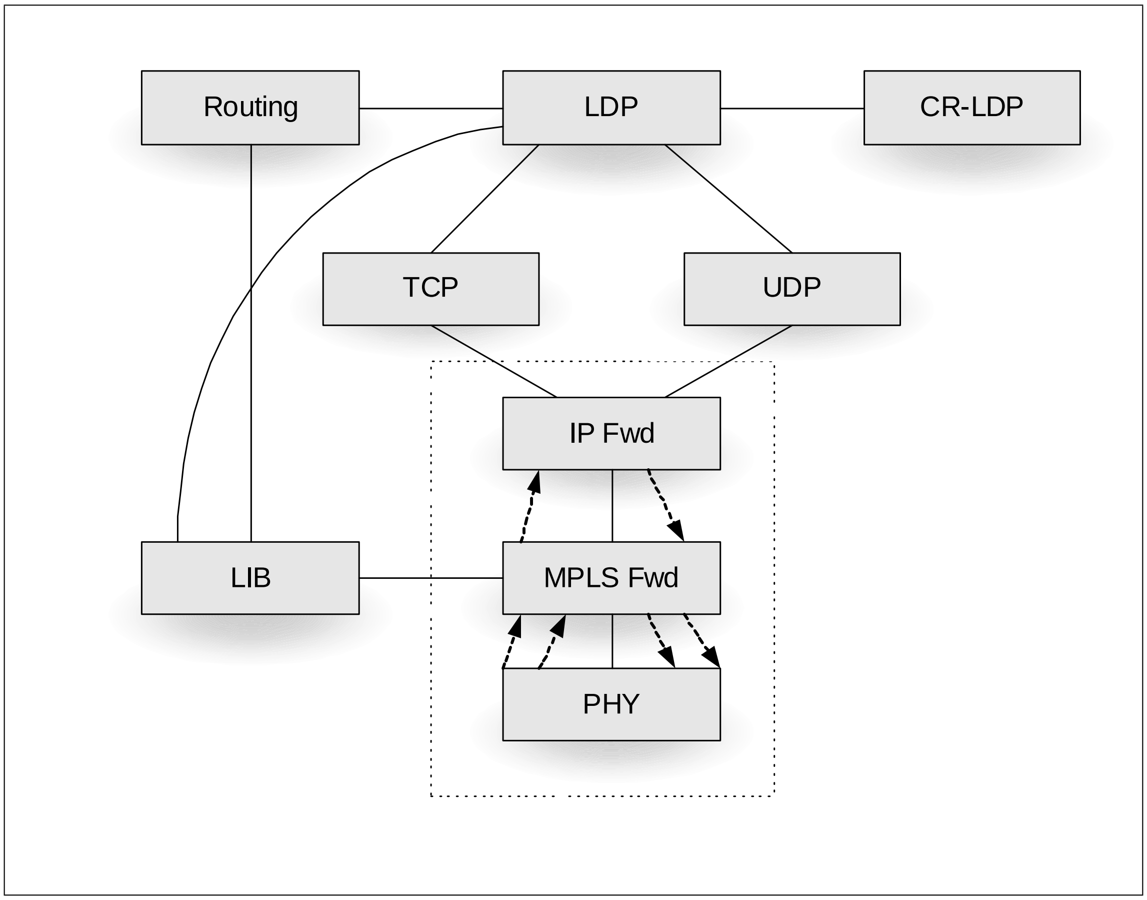


Figure 9: MPLS Protocol Stack

# 3.0 ADAPTIVE CAPACITY ALLOCATION

## 3.1 INTRODUCTION

In order to develop effective adaptive capacity adjustment schemes under a dynamic traffic scenario, we need to study the time varying behavior of the loss system model. Since determining closed form expressions for the general non-stationary behavior of queueing systems is extremely difficult, we adopt a numerical solution approach using a numerical methods based approximation technique. Specifically, we adopt the Pointwise Stationary Fluid Flow Approximation (PSFFA) method discussed in [48], [49]. In the following two sections, we derive the PSFFA model for the specific loss system under consideration followed by the adaptive capacity allocation scheme development for a single traffic. Then, we extend our work to address the case of multiple traffic classes.

## 3.2 A FLUID-FLOW MODEL FOR DYNAMIC TRAFFIC

A fluid-flow model can be applied at different levels and scenarios. In our case where we have a time-dependent, non-stationary loss system, the queue can be thought as a M(t)/M(t)/C(t)/C(t) type, and therefore, the connection level performance can be investigated. However, obtaining closed form expressions for the general non-stationary behavior of queueing systems is extremely difficult, so we adopt a numerical solution approach using a numerical methods based approximation technique. Specifically, we adopt the Pointwise Stationary Fluid Flow Approximation (PSFFA) method. This approximation has been firstly discussed in [48].

The PSFFA method models the average number in the system at a queue by one or more differential equations which are solved numerically. The use of the PSFFA approach to determine the nonstationary behavior of general finite and infinite capacity queueing systems is discussed in [38], [48], [49] and was shown to be reasonably accurate for the cases studied. Here, we derive the PSFFA model for the specific loss system under consideration. We define $x(t)$ as the state variable representing the average number of connections that are present on the link at time, $t$. Let $\dot{x}(t) = dx(t)/dt$ be the rate of change of the state variable with respect to time. From the flow conservation principle, the rate of change of the average number in the system is equal to the difference between the average arrival and departure rates. Let $f_{in}(t)$ and $f_{out}(t)$ denote the ensemble average number of connections *in* and *out* of the system at time $t$, respectively. Then the rate of change of the state variable can be related to the flow in and flow out by:

$$\dot{x}(t) = -f_{out}(t) + f_{in}(t) \tag{3.1}$$

This type of equation can be found in several places in the literature and is commonly referred to as a fluid flow or dynamic flow equation [33], [38], [48], [49]. To determine the particulars of the flow *in* and *out* of the system for the M/M/c/c queue case, we let $\pi_i(t)$; $i = 0; 1, ..., C(t)$ denote the state probabilities of the system with $\pi_i(t)$ representing the probability that there are $i$ connections in the system at time $t$. The flow into the system is just the offered load $\lambda(t)$ minus the portion of the offered load that is blocked. Hence we have:

$$f_{in}(t) = \lambda(t)(1 - \pi_{C(t)}(t)) \tag{3.2}$$

where $\pi_{C(t)}(t)$, denotes the probability that a connection is blocked at time $t$, given the current capacity $C(t)$. The flow out of the queue $f_{out}(t)$, is the current utilization of the $C(t)$ servers and is given by

$$f_{out}(t) = \mu\pi_1(t) + 2\mu\pi_2(t) + ... + C(t)\mu\pi_{C(t)}(t) \tag{3.3}$$

Where $\mu$ is the average service rate of a queue with exponential service time. This is shown in [37] (and reproduced in Appendix A) to be equivalent to:

$$f_{out}(t) = \mu x(t) \tag{3.4}$$

Thus, the fluid flow model becomes:

$$\dot{x}(t) = -\mu x(t) + \lambda(t)(1 - \pi_{C(t)}(t)) \tag{3.5}$$

Computing an exact solution for $\pi_{C(t)}(t)$ in the non-stationary case is extremely difficult and we use the PSFFA approach of approximating it from the steady state formulas for the system under study [48], [49]. For the M/M/C/C system we use the steady state functional relationships to estimate $\pi_{C(t)}(t)$ as a function of $x(t)$. At steady state, the average number of connections in the system $x$, the offered load $a$ Erlangs and the blocking probability $\pi_C$ are related by:

$$x = a[1 - \pi_C] \tag{3.6}$$

and

$$\pi_C = E(a, C) = \frac{\frac{a^C}{C!}}{\sum_{k=0}^{C} \frac{a^k}{k!}} \tag{3.7}$$

Note, that $\pi_C = E(a, C)$, where $E(a, C)$ is the well known Erlang-B loss formula. We assume the same functional relationships hold for the time varying behavior, and $x(t)$ and $C(t)$ values; we solve for $a(t)$and $\pi_{C(t)}(t)$ using the equations above. Specifically, equation (3.6) is rewritten as:

$$a(x(t)) = \frac{x(t)}{1 - E(a(x(t)), C(t))} \tag{3.8}$$

Since equation (3.7) is now a function of $C(t)$, and the offered load, $a(t)$ is a function of $x(t)$ , we get:

$$\pi_{C(t)}(t) = E(a(x(t), C(t)) = \frac{\frac{a(x(t))^{C(t)}}{C!}}{\sum_{k=0}^{C(t)} \frac{a(x(t))^k}{k!}} \tag{3.9}$$

Note that equation (3.8) is a fixed point equation which can be solved together with equation (3.9) in an iterative fashion to jointly determine $a(x(t))$ and $\pi_{C(t)}$. Thus, the PSFFA model of the system is given by (3.5) together with (3.8) and (3.9).

The PSFFA model can be solved numerically to determine the time varying behavior of the system as follows. We identify an initial condition for the state variable at time $t_0$ as $x(t_0)$ and an initial capacity value $C(t_0)$. The arrival rate is approximated by a constant over a small time step $\triangle t$, by $\lambda(t) = \lambda(t_0 + \triangle t/2)$ for $t \in [t_0, t_0 + \triangle t]$. Then the blocking probability $\pi_{C(t)}(t)$ is approximated by a constant over $t \in [t_0, t_0 + \triangle t]$ by solving (3.8) together with (3.9). The PSFFA model (3.5) can then be numerically integrated using a standard technique, such as, the Runge-Kutta-Fehlberg numerical method [34], where the step size parameter is optimized based on prescribed tolerance values. The numerical solution yields the value of the state variable at the end of the time interval, $x(t_0 + \triangle t)$, which then becomes the initial condition for the next time step $[t_0 + \triangle t, t_0 + 2 \triangle t]$. We then adjust the capacity and arrival rate for the new time step and the procedure is repeated for each time interval in the time horizon. The numerical solution technique can be written in algorithmic form to determine the behavior of the queue over time.

The performance of the PSFFA model and its accuracy in capturing the nonstationary behavior of the queuing systems can be evaluated by comparing the results with the ones obtained by integrating the associated Chapman-Kolmogorov differential equation (CK) model [37], [48]. In this context, the M/M/C/C Chapman-Kolmogorov model is determined and solved numerically. Defining $p_j(t)$ as the probability of $j$ connections being in the system at time $t$, $\mu$ as the mean service rate, and $\lambda(t)$ as the time varying mean arrival rate, $C$ as the fixed system capacity, the Chapman-Kolmogorov differential equations (CK model) are given by:

$$
\begin{aligned}
dp_0(t)/dt &= -\lambda(t)p_0(t) + \mu p_1(t) \\
dp_j(t)/dt &= \lambda(t)p_{j-1}(t) - (\lambda(t) + j\mu)p_j(t) \\
&\qquad + (j+1)\mu p_{j+1}(t), \qquad\qquad 0 < j < C \\
dp_C(t)/dt &= \lambda(t)p_{C-1}(t) - C\mu p_C(t)
\end{aligned}
$$

This differential equation model can be solved numerically in a fashion similar to the numerical solution of the PSFFA model. One approximates the arrival rate by a constant over a small time step and applies a standard numerical integration algorithm to solve the differential equations over the time step. This procedure is repeated over the time horizon as detailed in [48]. From the solution to the CK model for the time dependent state probabilities $p_j(t)$, one can directly determine the time varying mean number in the system using $x(t) = \sum_{i=0}^{N} i\ p_i(t)$ and the blocking probability using $\pi_C(t)$.

Figures (10) through (13) illustrate typical transient behavior of the average number in the system and the blocking probability of the loss model using the PSFFA and CK models with a stationary and non-stationary offered load respectively. The models are for the case of M(t)/M/24/24 with mean rate $\mu = 1$ and traffic load $a(t) = \lambda(t)/\mu = 15$ for the stationary load and $a(t) = 15 + 3\sin(0.1(t + 20))$ for the nonstationary case. From the figures, one can see that the PSFFA model is quite accurate in tracking both the mean number in the system and the blocking probability.

### 3.3 ADAPTIVE CAPACITY ALLOCATION FOR SINGLE TRAFFIC

The goal of implementing the bandwidth adjustment scheme is to show that it can provide a mechanism which will allow for provisioning of the QoS requirements. In our case, the system performance measure of interest is the connection blocking, $b(t) = \pi_C(t)$. Hence, we consider a non-stationary offered load to a single-link, loss system. Single link loss systems are typically used to represent a single circuit switched link, a virtual path in an ATM network or a label-switched path in a MPLS network carrying homogeneous traffic. We consider the case where the link capacity $C(t)$ is time varying and is counted as a multiple

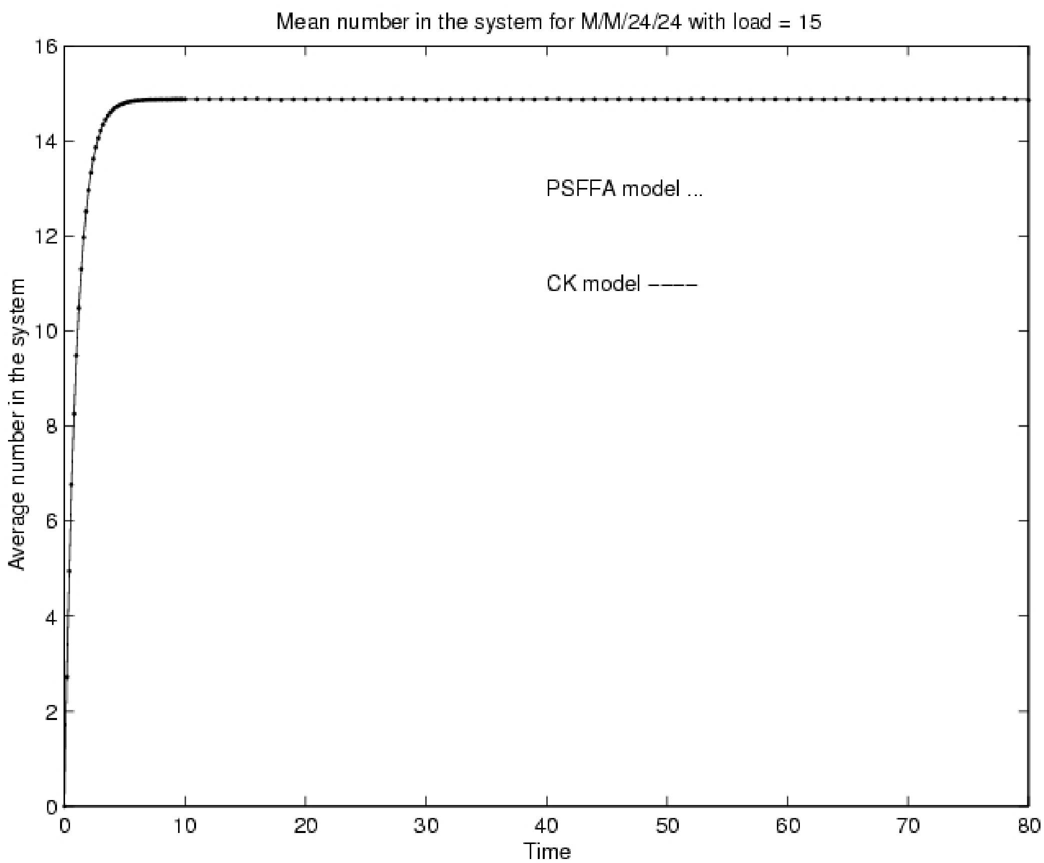


Figure 10: Mean number of connections in the system (stationary traffic): PSFFA comparison with CK Model.

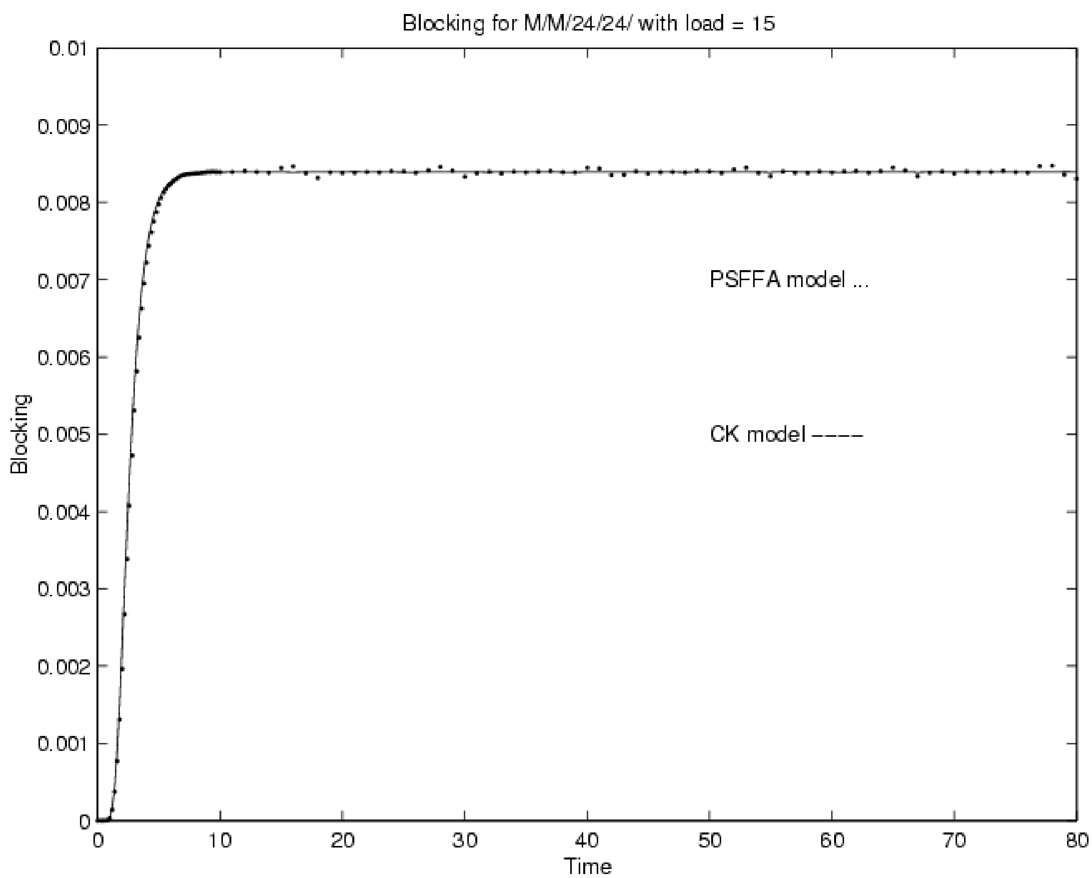


Figure 11: Blocking probability in the system (stationary traffic): PSFFA comparison with CK Model.

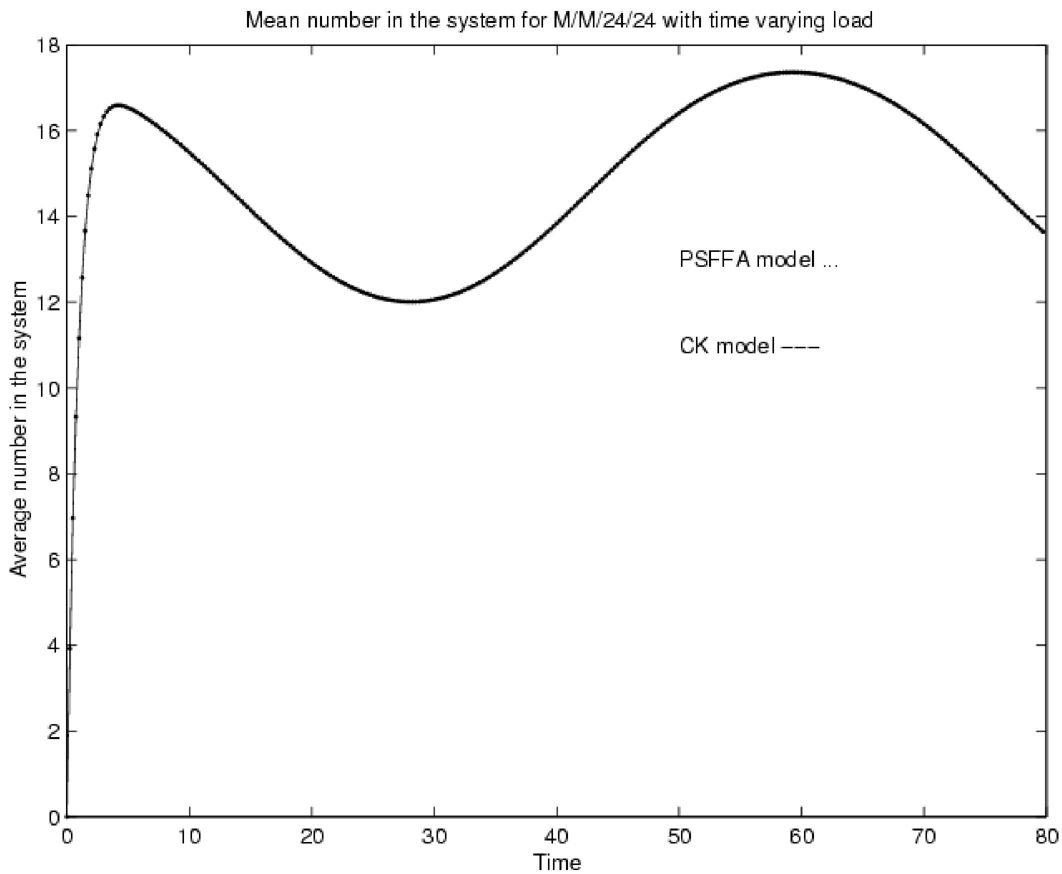


Figure 12: Mean number of connections in the systems (non-stationary traffic): PSFFA comparison with CK model.

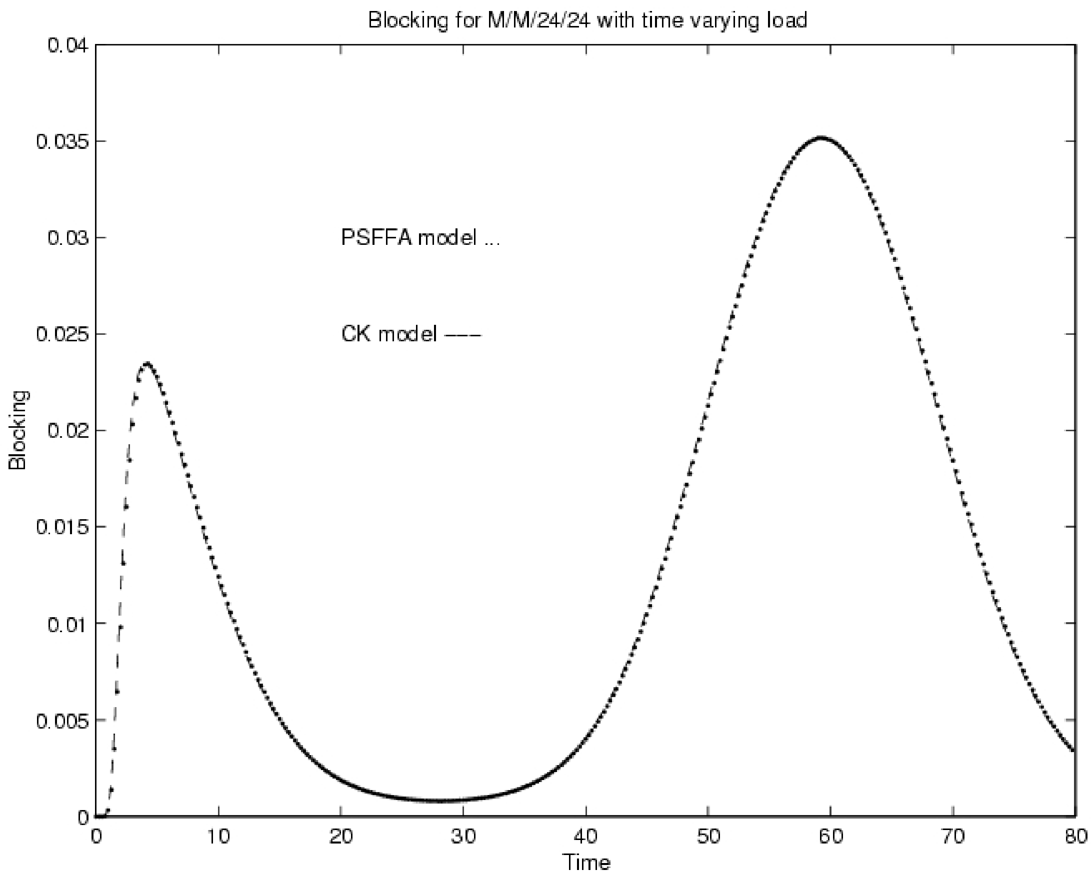


Figure 13: Blocking Probability in the system (non-stationary traffic): PSFFA Comparison with CK Model.

of a basic bandwidth unit (i.e., $C(t)$ takes on positive integer values). We assume that that the time varying offered load at time $t$ is given by $a(t)$ Erlangs. It is characterized by a nonstationary Poisson process with mean arrival rate $\lambda(t)$ at time $t$. This is consistent with the measurement results and theoretical models reported in [35], [39], [33], [46], [38]. It may be noted that work based on actual measurements from data traffic has reported that while a Poisson model is not applicable at the packet level, a time-varying Poisson model is applicable at connection or session level [43]. Since in this paper, our interest is only at the connection-level performance, the use of time-varying Poisson arrival is appropriate. The connection holding (duration) time is assumed to be exponentially-distributed, with mean, $\mu$. Thus, the offered load, the arrival rate and the holding time are related by $a(t) = \lambda(t)/\mu$. Note that there is no restriction on the arrival pattern of traffic within a connection, only on the holding time of connections and the time between connection requests. A connection arrival, at time, $t$, finds capacity $C(t)$, if there is a free unit of capacity, in the present value of $C(t)$, to accommodate the connection, then the connection is accepted and uses a unit of bandwidth, otherwise, the connection is blocked and cleared. Note that $C(t)$ is time-dependent and represents the adjustable capacity in the network link.

The developed scheme as shown in Figure (14) is based on the idea that the blocking rate should be maintained within a desired range, which corresponds to the QoS requirements. In other words, if the connection blocking rate is beyond an acceptable level, a request for additional capacity is desirable to provide an acceptable level of blocking. By the same token, if the current blocking is significantly lower than what is acceptable, it may indicate that this service class may have too much idle bandwidth, thus it is desirable to release some of it. Here, we apply a Lyapunov based control using Lyapunov stability theory to design an adaptive bandwidth controller that has the advantage of keeping the trajectory of a QoS metric (blocking rate in our case) within a error bound as a desired value [45]. Lyapunov control theory is usually deployed when a system parameter is desired to be maintained between two thresholds. As mentioned earlier, the blocking probability is our measure of interest. Since the blocking rate, $\pi_C(t)$, is related to the number of connections in the system, $x(t)$, as indicated in equation (3.6), we could control the blocking rate by controlling the number of connections. The desired value of the blocking rate is translated

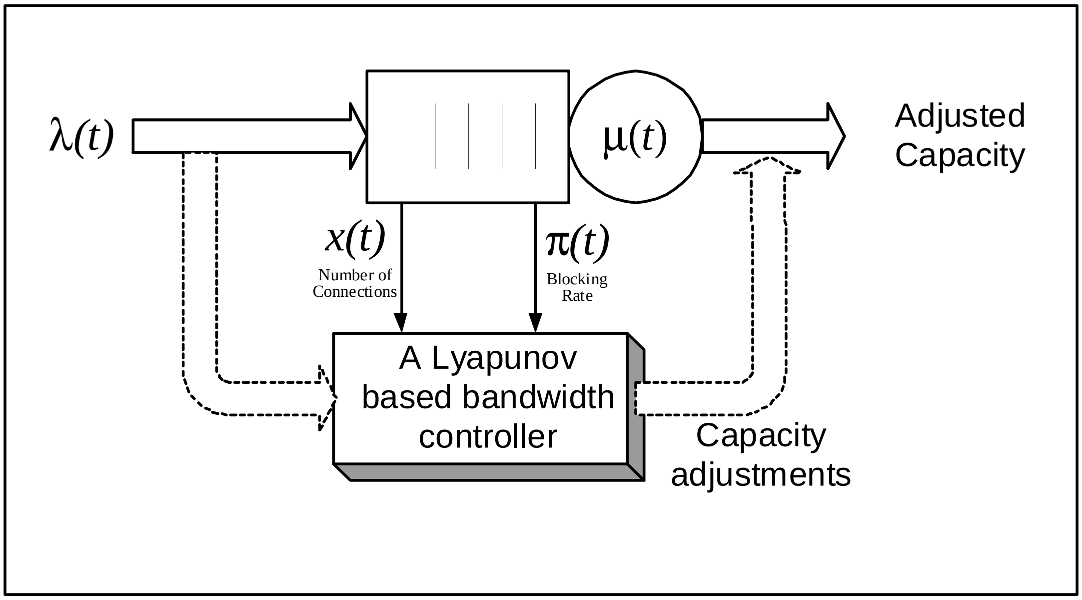


Figure 14: Our model for adaptive capacity allocation

to a corresponding value for the number of connections. Using this parameter translation simplifies the computation difficulties of the differential equations. Now, we present the Lyapunov theorem for stability [40] and then we show its application in determining the controller design:

**Theorem 1.** *The equilibrium of equation* $\dot{e} = f(t, e)$ *at time* $t_0$ *is stable if there exists a continuously differentiable positive definite function,* $V(t, e)$*, such that* $\dot{V}(t, e) < 0$*,* $\forall t \geq t_0$*,* $\forall e \in B_r$*, where* $B_r = \{e : \|e\| \leq r\}$*,* $r > 0$*.*

In theorem 1, the function $V(t, e)$ is called the Lyapunov function and satisfaction of the theorem guarantees that $e(t)$ will approach its equilibrium at an exponential rate of decay. Proof of the stability theorem and its variations for asymptotic stability are given in [39]. Here we use the stability theorem to craft a control that will guarantee stability of the number of connection, $x(t)$. Note that the blocking probability, $\pi_C(t)$, is related to the number of connections in the system, $x(t)$, by the relation of $x = a[1 - \pi_C]$ in the steady state. We assume this relationship holds under non-stationary conditions as well. Hence, we translate the value of the desired blocking probability to corresponding value of number of connections. Then, the target is to maintain the number of connections within a certain range of the desired number of connections, which results in satisfying the QoS requirements.

Let $\pi_C^d$ denote the desired blocking rate, which maps into $x_d$ as the corresponding desired number of connections in the system.

The error then can be defined as:

$$e = x_d - x \tag{3.10}$$

Taking the derivative of both sides yields,

$$\dot{e} = \dot{x}_d - \dot{x} \tag{3.11}$$

Substituting (3.5) in the above equation yields,

$$\dot{e} = \dot{x}_d - (-\mu x + \lambda(1 - \pi_C^d)) \tag{3.12}$$

For the above equation to be stable, we must determine a Lyapunov function such that $V(x) \geq 0$, and its derivative is negative, $\dot{V}(x) < 0$. Here, wedefine Lyapunov function as:

$$V = \frac{1}{2}e^2 \tag{3.13}$$

which is greater than or equal zero. Then, the derivative of $V(x)$ is,

$$\dot{V} = e\dot{e} = e(\dot{x}_d - (-\mu x + \lambda(1 - \pi_C^d)) \tag{3.14}$$

For stability,

$$e(\dot{x}_d - (-\mu x + \lambda(1 - \pi_C^d)) < 0 \tag{3.15}$$

or,

$$\dot{x}_d + \mu x - \lambda + \lambda \pi_C^d < 0 \tag{3.16}$$

Thus,

$$\pi_C^d < -\frac{\mu}{\lambda} x - \frac{\dot{x}_d}{\lambda} + 1 \tag{3.17}$$

The desired blocking probability can be evaluated since all variables on the RHS of (3.17) are known. The variable, $\dot{x}_d$, is determined using (3.11), i.e., $\dot{x}_d = \dot{x} + \dot{e}$, where $\dot{x}$ is given in Eq. (3.5), and $\dot{e}$ is approximated as $\dot{e} = (e_{t-\triangle t} - e_t)/\triangle t$. By substituting the bound on

$\pi_C^d$ in (3.7), i.e., $\pi_C^d = \frac{a^C}{C!} / \sum_{k=0}^{C} \frac{a^k}{k!}$, and solving for $C$, we can get the proper amount of capacity adjustment that guarantees the stability of the system for the desired call blocking rate, and hence, the QoS requirement is met.

This formulation along with the numerical solution can be written in an algorithmic form for solution over a time interval $(t_0, t_f)$ as follows:

*Algorithm 1: Adaptive Capacity Allocation for Single Traffic Class:*

1. Initialization: set current time $t$, to $t = t_o$ establish the initial system occupancy $x(t) = x(t_0)$, system capacity $C(t) = C(t_0)$ and specify a time step $\triangle t$.

2. Approximate the offered load $a(t)$ by a constant $a$ over $[t; t + \triangle t]$ with $a = a(t + \triangle t/2)$.

3. Approximate $\pi_{C(t)}(t)$ over $[t; t + \triangle t]$ by a constant $\pi_{C(t)}$ by solving (3.8) and (3.9) iteratively until the change in $a(x(t))$ does not exceed a prespecified $\epsilon$.

4. Utilizing $x(t)$, $a$ (from step 2), and $\pi_{C(t)}$ (from step 3), numerically solve the differential equation given by (3.5) over the small time interval $\triangle t$ using a standard technique (e.g. Runge-Kutta), and get the new system occupancy at time $t + \triangle t$; $x(t + \triangle t)$.

5. Calculate the desired blocking rate, $\pi_C^d$, using (3.17). Solve for the required capacity, C, from (3.7), and update the link bandwidth.

6. Increment time, $t = t + \triangle t$. If $t < t_f$, go to 2, else stop.

## 3.4 ADAPTIVE CAPACITY ALLOCATION FOR MULTIPLE CLASS TRAFFIC

For the case of multiple classes, two scenarios are to be considered; first, when each traffic class is designed to have a separate capacity (coupling between traffic classes is ignored), and the second one is when the link capacity is shared and all traffics have an access to the available bandwidth (traffic classes are coupled). In terms of equations, we express the first case (separate capacity), which is the rate of change of each traffic is a function of the number of connection of its traffic and blocking rate of that specific class as:

$$\begin{aligned} \dot{x}_1(t) &= f(x_1(t), \pi_1(t)) \\ \dot{x}_2(t) &= f(x_2(t), \pi_2(t)) \\ &\vdots \\ \dot{x}_k(t) &= f(x_k(t), \pi_k(t)) \end{aligned} \tag{3.18}$$

While in the second case (shared capacity), the rate of change of the state variable is function of the number of connections and blocking rate of all traffic classes, and that can be expresses as:

$$\begin{aligned} \dot{x}_1(t) &= f((x_1(t), \pi_1(t)), (x_2(t), \pi_2(t)), ..., (x_k(t), \pi_k(t)) \\ \dot{x}_2(t) &= f((x_1(t), \pi_1(t)), (x_2(t), \pi_2(t)), ..., (x_k(t), \pi_k(t)) \\ &\vdots \\ \dot{x}_k(t) &= f((x_1(t), \pi_1(t)), (x_2(t), \pi_2(t)), ..., (x_k(t), \pi_k(t)) \end{aligned} \tag{3.19}$$

These two cases will be further investigated in the next two subsections then followed by a comparison between them.

### 3.4.1 Adaptive Capacity Allocation for Multiple Classes with no coupling

When each traffic class has a separate allocated bandwidth, the scheme for single traffic developed previously can be directly applied to each traffic class as shown in Figure (15).

In this case, each incoming traffic has an allocated capacity, $C_i$ , $i = 1, 2, ..., n$ for $n$ service classes. The capacity, $C_i$, is dynamically adjusted in order to meet the QoS. One condition

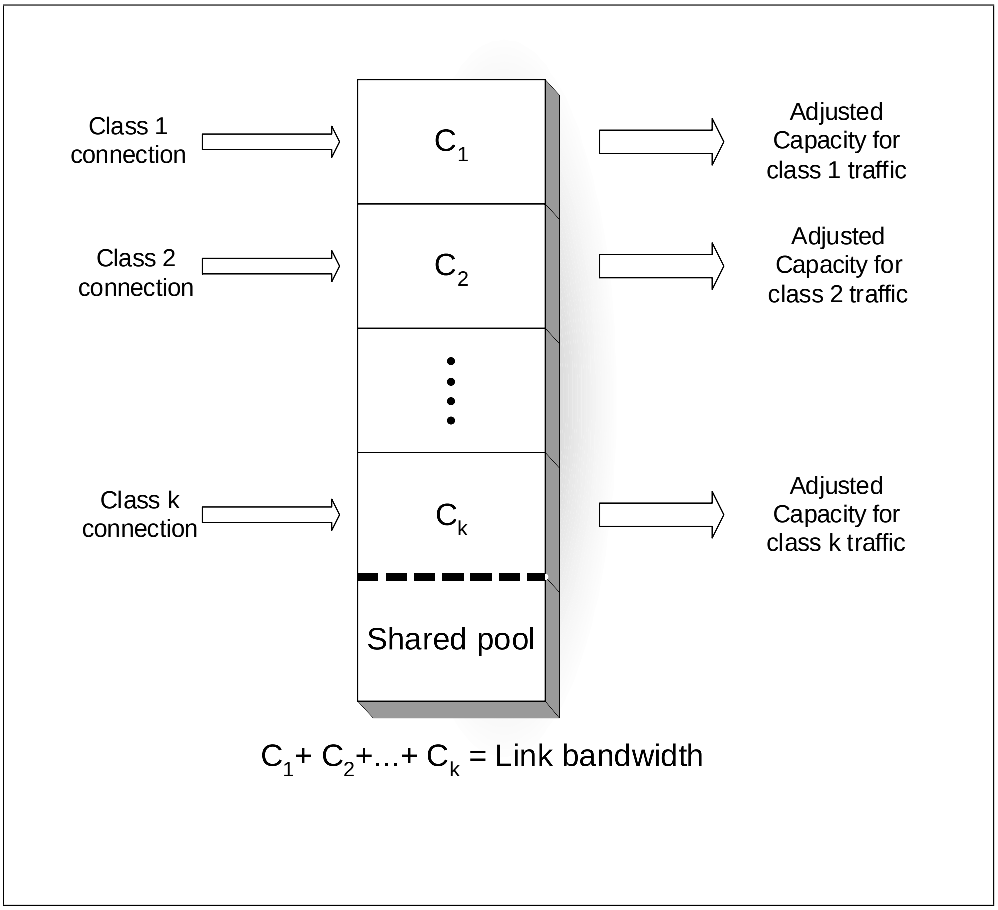


Figure 15: A General Access Senario for Mutiple Traffic Classes with no coupling

that has to be met is that the total sum of capacities should not exceed the physical link bandwidth (i.e., $\sum_{i=1}^{n} C_i \;\leq\; Physical\ Link\ Bandwidth$), otherwise, one or more classes may not receive the required QoS. When the bandwidth requirement of a certain service class increases, additional capacity may be requested from a shared pool capacity, and vise versa, when the capacity is excessive, some of it may be deallocated and assigned to the shared pool capacity for use by other services classes as long as the QoS is being met. The simulations and analysis of this case will be discussed further and illustrated in the next chapter.

### 3.4.2 Adaptive Capacity Allocation for Multiple Classes (Shared Capacity)

In this case, we continue with the same assumptions for the link that the capacity is counted as a multiple of basic bandwidth units (BBU), and therefore, the link capacity has $C$ units of BBUs. The link is subject to traffic from $k$ different service classes as shown in Figure (16). Each class $i$ connection demands simultaneous access to any $m_i$ units of the BBUs for the duration of the connection. Note that $m_i$ is assumed to be constant during the connection and is already defined in the Service Level Agreement (SLA) within the MPLS network. Again, we assume for the traffic arrival process as a Poisson process with mean rate $\lambda_i(t)$ and the connection holding time as exponentially distributed with mean $1/\mu_i$ for each traffic class $i$.

To derive our scheme of adaptive capacity allocation, we begin with developing a unified model for nonstationary system from which we can determine the blocking probability and number of connections in the system. Let $\overline{n}_i(t)$ be the number of class $i$ connections in the system at time $t$. The the state of the stochastic process describing the system is the vector-valued process $\overline{n}(t) = (\overline{n}_1(t), \overline{n}_2(t), ..., \overline{n}_k(t))$. Under the assumption above, the state vector $\overline{n}$ defines a $k$ dimensional, continuous-time Markov chain. Let $\Omega$ denote the finite state space of $\overline{n}$ and $P(t; n_1, n_2, ..., n_k) = P(\overline{n}_1(t) = n_1, \overline{n}_2(t) = n_2, ..., \overline{n}_k(t) = n_k)$, denote the probability that the system is in state $(n_1,\ n_2, ...,\ n_k)$ at time $t$. The size of the finite state space, $\Omega$, is determined by constraints $n_i(t) \geqslant 0$, $\forall i$, at time $t$ and $0 \leqslant \sum_{i=1}^{k} n_i(t)$ $m_i \leqslant C$, at time $t$.

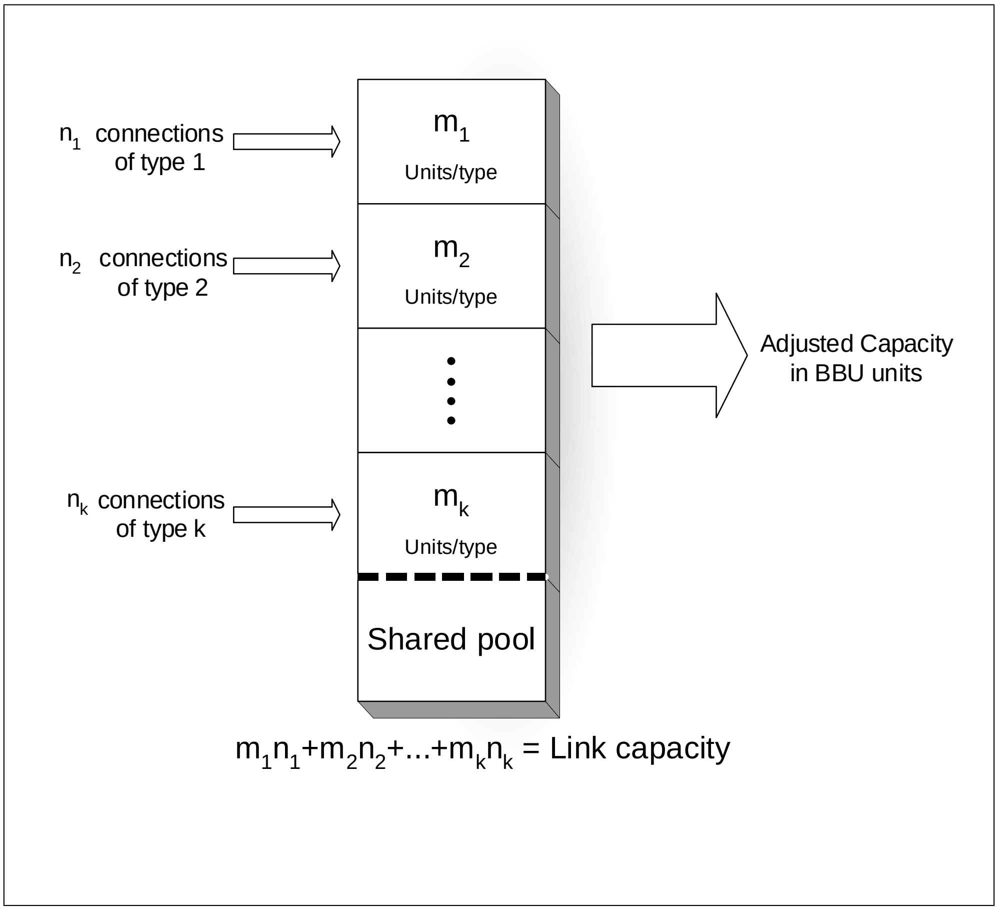


Figure 16: A General Access Senario for Multiple Classes with Shared Capacity

In order to illustrate that, we first analyze the case of two traffic class ($k = 2$), then we elaborate our discussion to include the case of which $k > 2$.

An illustration of the finite state space $\Omega$ and state transitions of the shared capacity scheme is shown in Figure (17). In case of two classes, the model has two dimensional Markov process $(\overline{n}_1(t), \overline{n}_2(t))$.

As shown in Figure (17), the state space has the following form:

$$\Omega = \left\{ \begin{array}{ccccccc} (0,0) & (1,0) & (2,0) & \dots & \dots & \dots & (I_0,0) \\ (0,1) & (1,1) & (2,1) & \dots & \dots & \dots & (I_1,1) \\ (0,2) & (1,2) & (2,2) & \dots & \dots & \dots & (I_2,2) \\ : & : & : & : & : & : & : \\ (0,n_2) & (1,n_2) & (2,n_2) & \dots & (n_1,n_2) & \dots & (I_{n_2},n_2) \\ : & : & : & : & : & : & : \\ (0,J_0) & (1,J_0) & (2,J_0) & \dots & \dots & \dots & (I_{J_0},J_0) \end{array} \right\} \tag{3.20}$$

Where, $J_0 = \lfloor C/m_2 \rfloor$ and $I_{n_2} = \lfloor (C - n_2 m_2)/m_1 \rfloor$, $n_2 = 0, 1, 2, ..., J_0$. Notice that each row of $\Omega$ can contain a different number of entries depending on the state space determined by the specific parameters of the traffic. From (3.20), the size $K$ of the state space $\Omega$ is given by:

$$K = \sum_{n_2=0}^{J_0} (I_{n_2} + 1) = J_0 + 1 + \sum_{n_2=0}^{J_0} I_{n_2} \tag{3.21}$$

The feasible region of $\Omega$ can be described by introducing a new function called an indicator function, $\chi(n_1, n_2)$, which can be in expressed as:

$$\chi(n_1, n_2) = \begin{cases} 0, \text{if } (n_1, n_2) \notin \Omega \\ 1, \text{if } (n_1, n_2) \in \Omega \end{cases} \tag{3.22}$$

Here, the indicator function is used to enable state transitions among feasible states. In other words, if $(n_1, n_2)$ is a feasible point in the state space $\Omega$, we can determine whether neighboring states, $(n_1 - 1, n_2)$, $(n_1, n_2 - 1)$, $(n_1 + 1, n_2)$, $(n_1, n_2 + 1)$ are in $\Omega$ by comparing

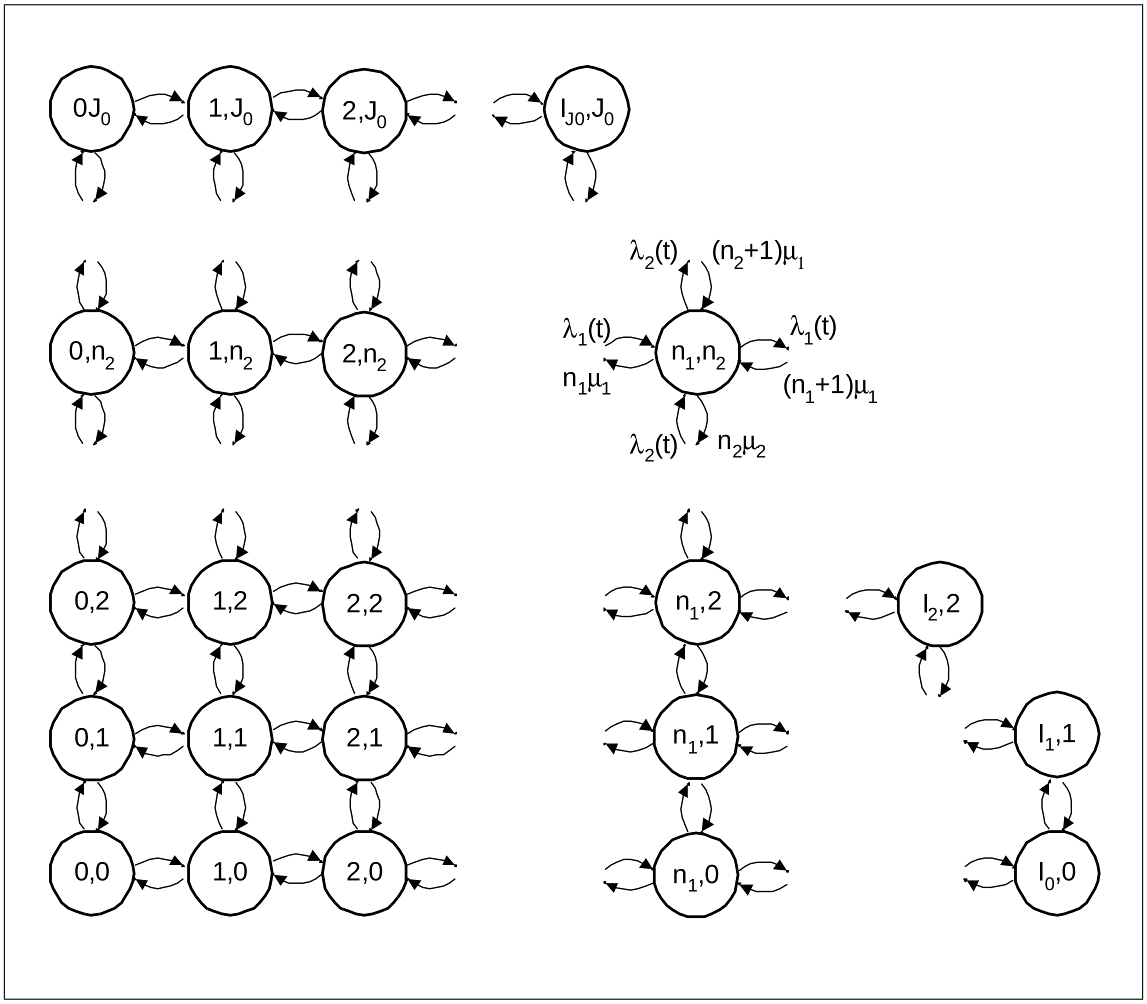


Figure 17: General state transitions diagram for two traffic classes (k=2)

them to the boundary points of $\Omega$ defined by $(I_{n_2}, n_2)$. Also, we define a set of acceptance functions, $\alpha_i(n_1, n_2)$, that determine whether a service class arriving to the system in state $(n_1, n_2)$ is accepted. The acceptance function is defined as:

$$\alpha_i(n_1, n_2, ..., n_k) = \begin{cases} 0, \text{ if } m_i > C - \sum_{j=1}^{k} n_j m_j \\ 1, \text{ if } m_i \leqslant C - \sum_{j=1}^{k} n_j m_j \end{cases} \tag{3.23}$$

Utilizing the indicator and acceptance functions, the Chapman-Kolmogorov differential equations describing the time varying behavior of the Markov process $(\overline{n}_1(t), \overline{n}_2(t))$ can be derived. So consider an arbitrary state $(n_1, n_2) \in \Omega$ : The rate of change in the state probability $\dot{P}(t; n_1, n_2)$ is the difference between the flow into the state and the flow out of the state, resulting in the following equation:

$$\begin{aligned} \tfrac{dP(t;n_1,n_2)}{dt} &= \lambda_1(t)\alpha_1(n_1-1, n_2)\chi(n_1-1, n_2)P(t; n_1-1, n_2) \\ &+\lambda_2(t)\alpha_2(n_1, n_2-1)\chi(n_1, n_2-1)P(t; n_1, n_2-1) \\ &+(n_1+1)\mu_1\chi(n_1+1, n_2)P(t; n_1+1, n_2) \\ &+(n_2+1)\mu_2\chi(n_1, n_2+1)P(t; n_1, n_2+1) \\ &-(\lambda_1(t)\alpha_1(n_1, n_2) + \lambda_2(t)\alpha_2(n_1, n_2) + n_1\mu_1 + n_2\mu_2)* \\ &P(t; n_1, n_2) \end{aligned} \tag{3.24}$$

The first four terms on the right hand side of (3.24) represent the flow into state $(n_1, n_2)$ from adjacent states, whereas the last term represents the flow out of $(n_1, n_2)$. With appropriate specification of the indicator and acceptance functions, equation (3.24) holds for all states $(n_1, n_2) \in \Omega$ resulting in a set of $K$ differential equations describing the system.

Following the same equation definitions above, the probability of blocking, , for each traffic class can be determined by noting that $Pb_i(t) = 1 - P(Class\ i\ connection\ is\ accepted)$, which results in the following relationship, where $\alpha_i(n_1, n_2)$ is the acceptance function for class $i$ in the case of two traffic classes and $P(t; n_1, n_2)$ is the probability that there are $n_1$ type 1 connections and $n_2$ type 2 connections in the system:

$$Pb_i(t) = 1 - \sum_{(n_1,n_2)\in\Omega} \alpha_i(n_1, n_2)P(t; n_1, n_2), \ \forall i \tag{3.25}$$

The analysis above can be extended easily to the general k-dimensional case (i.e., for $k > 2$). For this case, the state space of the shared capacity scenario has the following form:

$$\Omega = \left\{ (n_1, n_2, ..., n_k) \mid 0 \leq n_i \leq I_i,\ i = 1,\ 2, ...,\ k;\ \sum_{i=1}^{k} n_i\ m_i \leq C \right\} \tag{3.26}$$

where, $I_i = \lfloor C/m_i \rfloor$, $i = 1, 2, ..., k$. As before, an indicator function can be defined to express the feasible region of $\Omega$:

$$\chi(n_1, n_2, ..., n_k) = \begin{cases} 0,\ \text{if } (n_1, n_2, ..., n_k) \notin \Omega \\ 1,\ \text{if } (n_1, n_2, ..., n_k) \in \Omega \end{cases} \tag{3.27}$$

Also, the acceptance function, $\alpha_i(n_1, n_2, ..., n_k)$, is defined the same way mentioned earlier in (3.23). Then we can write the general Chapman-Kolmogorov differential equation for $\overline{n} \in \Omega$ as follows:

$$\begin{aligned} \tfrac{dP(t;n_1,n_2,...,n_k)}{dt} \quad &= \textstyle\sum_{i=1}^{k} \lambda_i(t)\alpha_i(n_1, ..., n_{i-1}, n_i - 1, n_{i+1}, ..., n_k)* \\ &\chi(n_1, ..., n_{i-1}, n_i - 1, n_{i+1}, ..., n_k)* \\ &P(t; n_1, ..., n_{i-1}, n_i - 1, n_{i+1}, ..., n_k) \\ &+ \textstyle\sum_{i=1}^{k}(n_i + 1)\mu_i\chi(n_1, ..., n_{i-1}, n_i + 1, n_{i+1}, ..., n_k)* \\ &P(t; n_1, ..., n_{i-1}, n_i + 1, n_{i+1}, ..., n_k) \\ &- \textstyle\sum_{i=1}^{k}(\lambda_i(t)\alpha_i(n_1, n_2, ..., n_k) + n_i\mu_i)P(t; n_1, n_2, ..., n_k). \end{aligned} \tag{3.28}$$

The first terms on the right hand side of (3.28) represent the flow in from adjacent states with one less customer (immediately below in state space), whereas, the second term represents flow in from adjacent states with one more customer. The last term sums the flow out of the current state.

**Example 1:** *If we assume that we have a link capacity of $C = 8$ BBUs for the case of two service classes, and $m_1 = 1$ and $m_2 = 2$ ( $m_i$ is the number of basic bandwidth units for traffic class $i$ ), then the state transitions of this scheme is as illustrated in Figure (18). With this setting, the parameters are: $J_0 = 4, I_0 = 8, I_1 = 6, I_2 = 4, I_3 = 2, I_4 = 0$. Figures (19) and (20) illustrate the subset of states for which class 1 traffic is accepted and class 2 traffic is accepted respectively when arriving to the system. The states under the dotted line* *represent a subset of the states for which a class k traffic is accepted when arriving to the system. For each figure, note that the sum probabilities that system at those states outside the dotted line is the blocking probability.*

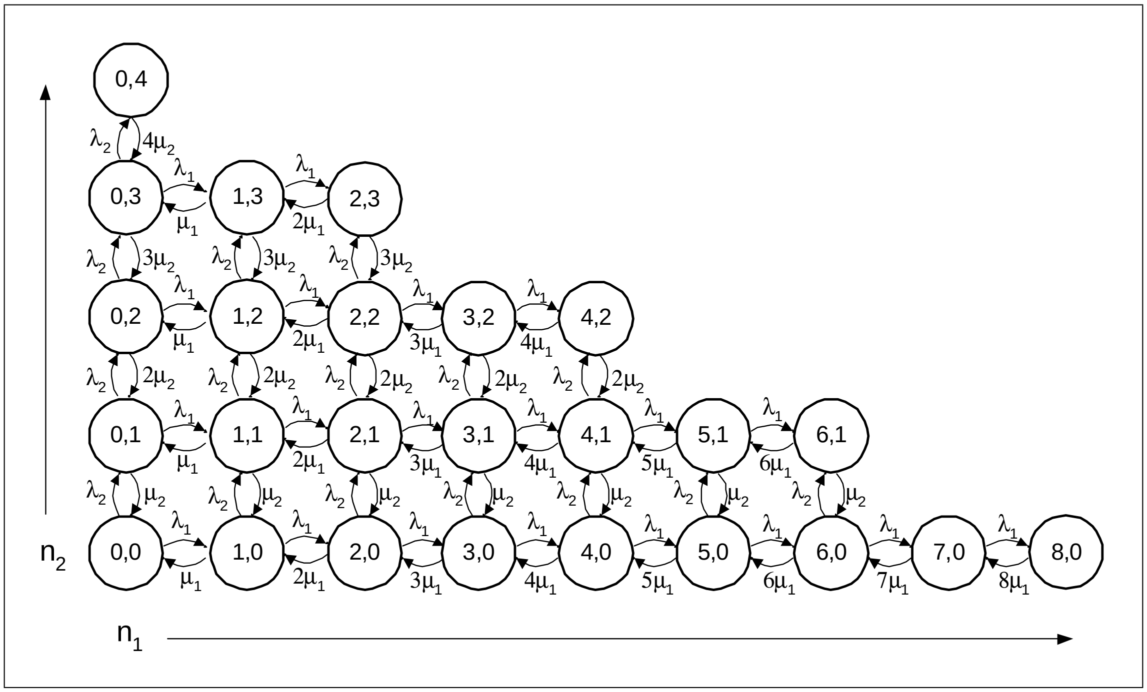


Figure 18: The state transition diagram with C=8, $m_1$=1, $m_2$=2.

Based on the discussion above, we propose a scheme for adaptive capacity allocation. First, we develop the scheme for the two traffic classes case as shown in Figures (17) and (18). Again, we adopt the same numerical methods solution used for the case of single traffic. By numerically solving (3.24), we can determine the total average number of connections in the system and that is given by:

$$\mathbf{x} = \sum_{n_2=0}^{J_0} \sum_{n_1=0}^{I_{n_2}} (n_1 + n_2)\ P(t; n_1, n_2), \text{ where } \mathbf{x} = x_1 + x_2 \tag{3.29}$$

This can also be written in terms of each traffic class as:

$$\begin{aligned} x_1 &= \textstyle\sum_{n_1} n_1 \sum_{n_2} P(t; n_1, n_2) \\ x_2 &= \textstyle\sum_{n_2} n_2 \sum_{n_1} P(t; n_1, n_2) \end{aligned} \tag{3.30}$$

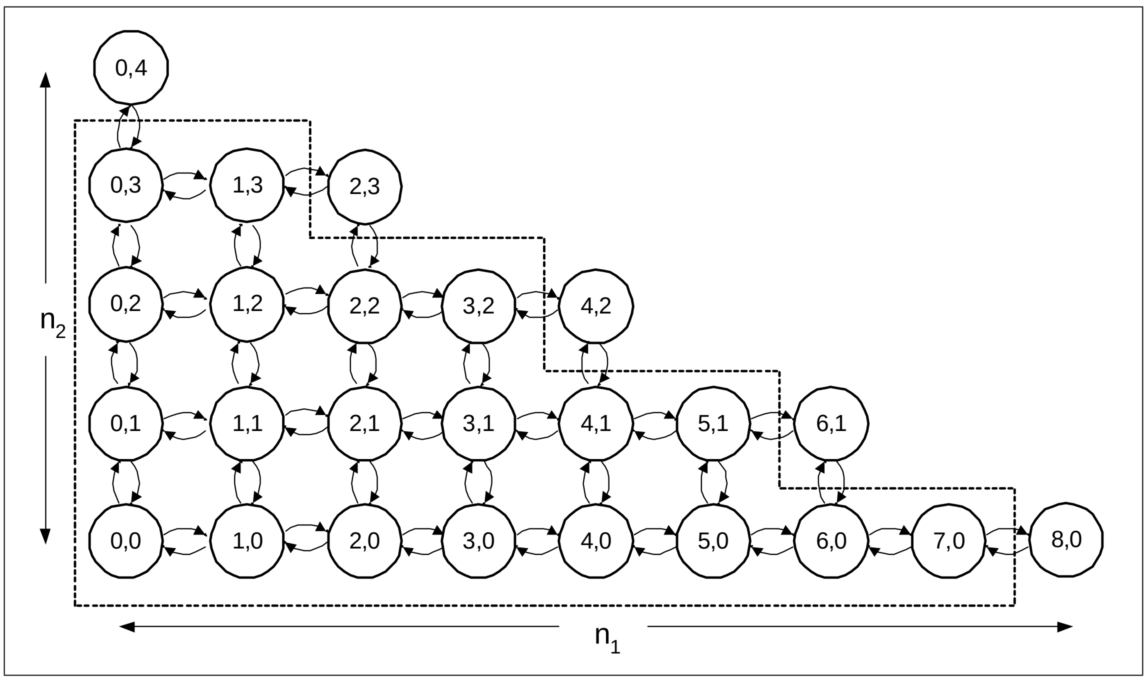


Figure 19: State diagram with C=8, $m_1$=1, $m_2$=2. Arriving class 1 traffic is accepted when the state is below the dotted line.

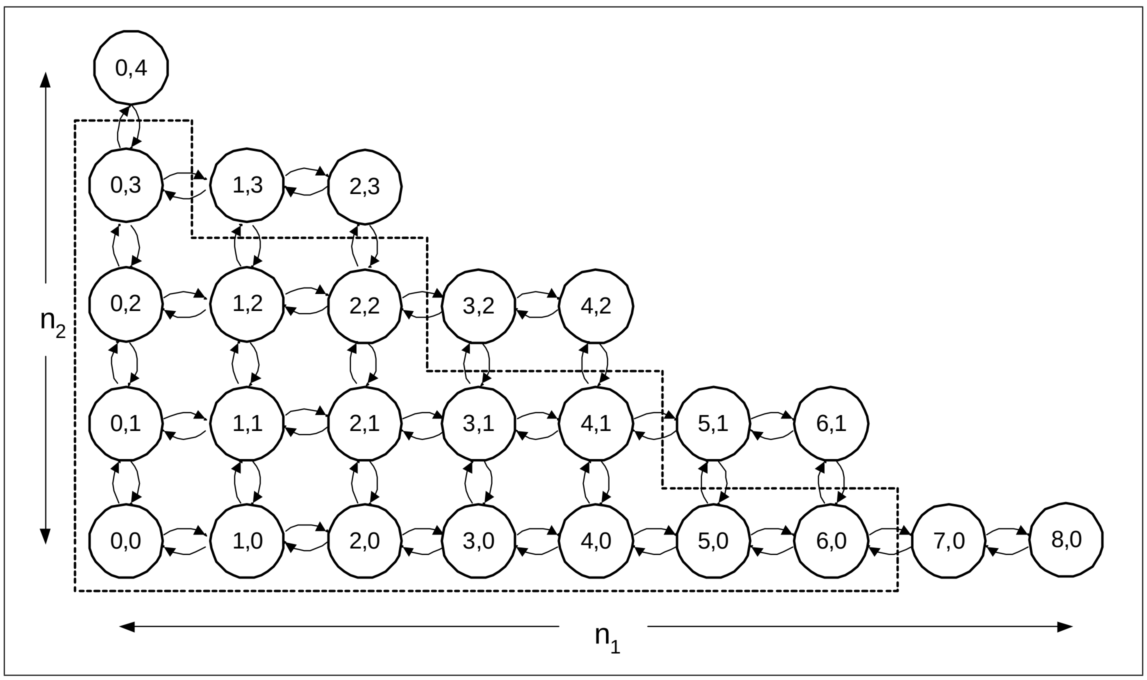


Figure 20: State diagram with C=8, $m_1$=1, $m_2$=2. Arriving class 2 traffic is accepted when the state is below the dotted line.

Thus, the rate of change of the mean number of connections is given by:

$$\dot{\mathbf{x}} = \sum_{n_2=0}^{J_0} \sum_{n_1=0}^{I_{n_2}} (n_1 + n_2) \; \dot{P}(t; n_1, n_2) \tag{3.31}$$

or,

$$\begin{aligned} \dot{x}_1 &= \textstyle\sum_{n_1} n_1 \sum_{n_2} \dot{P}(t; n_1, n_2) \\ \dot{x}_2 &= \textstyle\sum_{n_2} n_2 \sum_{n1} \dot{P}(t; n_1, n_2) \end{aligned} \tag{3.32}$$

Equation (3.31) is difficult to be solved for $\dot{P}$, so instead, we use an approximation by adopting the equation of the fluid flow model in (3.5) to approximate the number of connections. For k=2, we get:

$$\begin{aligned} \dot{x}_1(t) &= -\mu_1 x_1(t) + \lambda_1(t)(1 - \pi_1(t)) \\ \dot{x}_2(t) &= -\mu_2 x_2(t) + \lambda_2(t)(1 - \pi_2(t)) \end{aligned} \tag{3.33}$$

where, $\pi_i(t)$ is given by (3.25).

To show the accuracy of this approximation, we numerically solve equation (3.24) and compare the results of the number of connections given in (3.30) with the ones given in equation (3.33) after integration. So if we use the same parameters mentioned in the example, that is, $C = 8, m_1 = 1, m_2 = 2,\ \mu_1 = 1, \mu_2 = 2, a_1(t) = 15 + 10\sin(0.2(t + 20)), a_2(t) = 1.25 + 0.1\sin(0.2(t + 20))$, we get the results as shown in Figures (21) and (22).

From Figures (21) and (22), one can see that the approximation curve is tracking the exact solution. Therefore, we can rely on equation (3.33) from now and on. In a similar way of the adaptive capacity allocation for a single traffic, we re-write the formulation of the Lyapunov based adaptive bandwidth control in terms of equation (3.32) as follows:

$$\begin{gathered} e_k = x_k^d - x_k \ , \qquad k = 1, 2. \\ \dot{e}_k = \dot{x}_k^d - \dot{x}_k \\ \text{We define Lyapunov function as: } V = \tfrac{1}{2}e_1^2 + \tfrac{1}{2}e_2^2 \\ \text{Then, } \dot{V} = e_1\dot{e}_1 + e_2\dot{e}_2 \\ \text{For stability, } \dot{V} \leq 0, \text{ which implies } e_1\dot{e}_1 + e_2\dot{e}_2 \leq 0 \end{gathered} \tag{3.34}$$

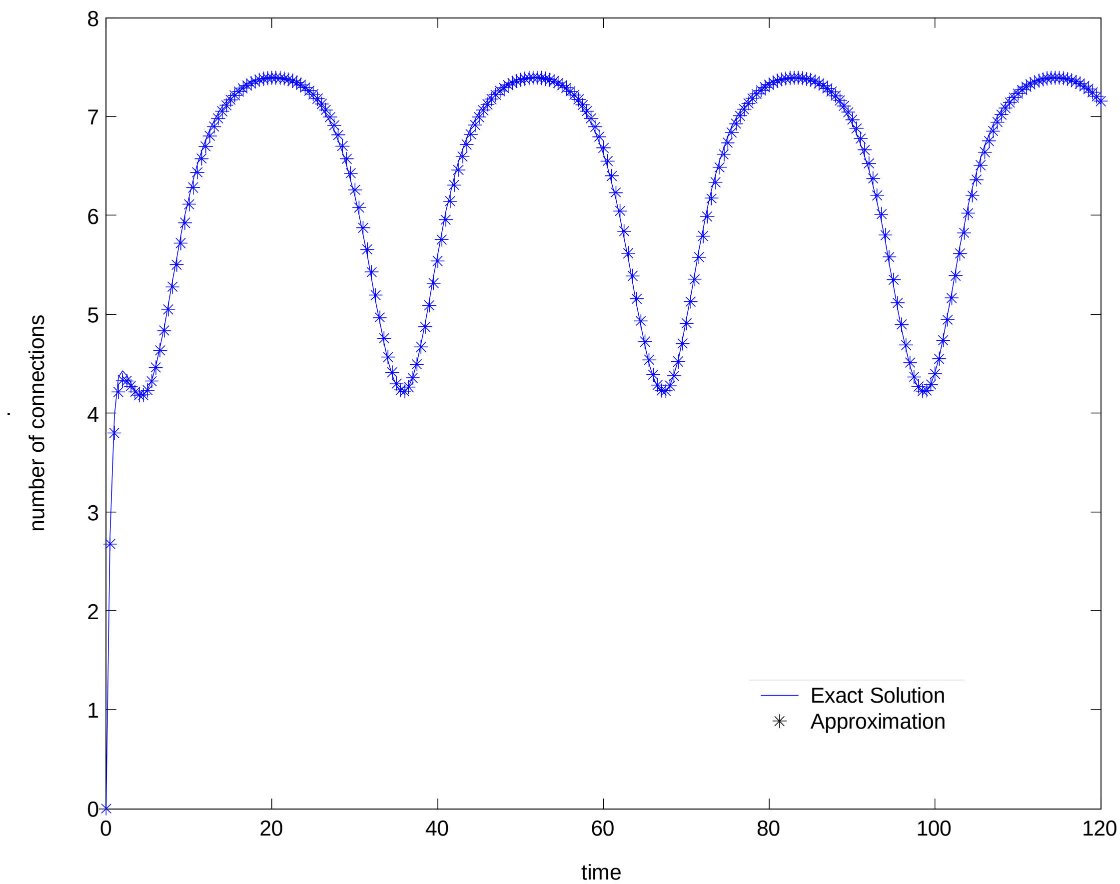


Figure 21: The number of connections for class 1: the exact solution comparison with the approximation one.

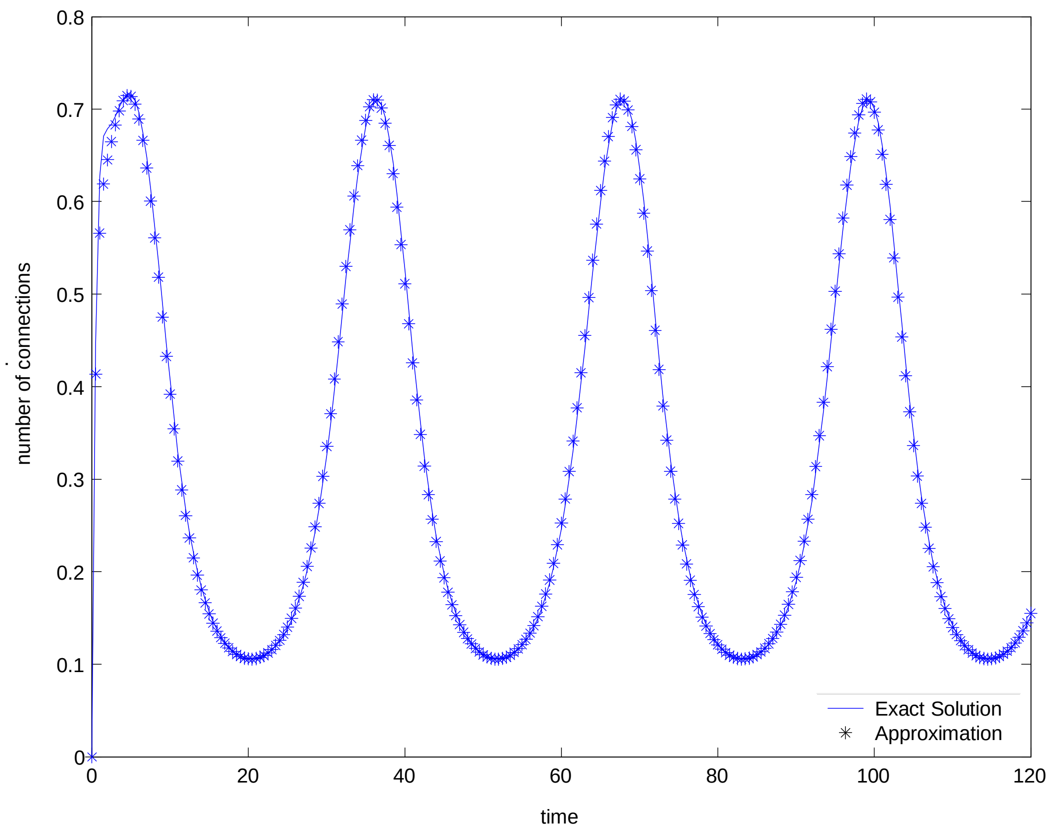


Figure 22: The number of connections of class 2: the exact solution comparison with the approximation.

Where $e$, as shown earlier, is the error between the desired mean number, $x^d$, and the actual mean number, $x$.For simplification, we assume that $e_1\dot{e_1} < 0$ and $e_2\dot{e_2} < 0$ is a possible solution.

Substituting (3.31) in (3.34) yields:

$$e_k\left(\dot{x}_k^d - \sum_{(n_1,n_2)\in\Omega}(n_1+n_2)\dot{P}(t;n_1,n_2)\right) \leq 0, \qquad \text{for } k=1,2 \tag{3.35}$$

Since equation (3.35) is difficult to solve, we use equation (3.33) instead since it gives an accurate approximation. So by substituting it in (3.34), we get:

$$e_k(\dot{x}_k^d - (-\mu_k x_k + \lambda_k(1-\pi_k^d(t))) \leq 0, \quad \text{for } k=1,2 \tag{3.36}$$

or,

$$\dot{x}_k^d + \mu_k x_k - \lambda_k - \lambda_k \pi_k^d(t) \leq 0, \qquad \text{for } k=1,2 \tag{3.37}$$

Thus,

$$\pi_k^d(t) \leq -\frac{\mu_k}{\lambda_k}x_k - \frac{\dot{x}_d}{\lambda_k} + 1, \qquad \text{for } k=1,2 \tag{3.38}$$

Again here, the blocking rate, $\pi_k^d(t)$, is a function of $J_0$, $I_{n_2}$, $N_{n_2}$,which are based on the link capacity as shown in equation (3.39) below (i.e., $J_0 = \lfloor C/m_2 \rfloor$ and $I_{n_2} = \lfloor (C - n_2 m_2)/m_1 \rfloor$, $n_2 = 0,1,2,...,J_0$, and $N_{n_2}$ is calculated using $I_{n_2}$ as: $N_{n_2} = N_{n_2} + I_{n_2} + 1$). Thus, given the value of $\pi_k^d(t)$, we calculate the corresponding capacity which gives us the required capacity adjustments as shown in the simulations. Here, we present the numerical algorithm for the adaptive capacity allocation for the k=2 case. Simulation results and outputs are demonstrated in the next chapter.

*Algorithm 2: Adaptive Capacity Allocation for Multiple Traffic Classes (k=2).*

*i) Determine* $J_0, I_0, I_1, I_2, ..., I_{J_0}$. *(Note:* $I_o \geq I_1 \geq I_2, ... \geq I_{n_n} \geq ... \geq I_{J_0}$*)*

$J_0 = \lfloor \frac{C}{m_2} \rfloor$;

$I_0 = \lfloor \frac{C-n_2m_2}{m_1} \rfloor$;

$n_1 = I_0;$

*for* $n_2 = 1$ to $J_0$

*while* $(n_1 \geq 0)$ *and* $(n_1 m_1 + n_2 m_2 > C)$

$n_1 = n_1 - 1;$

*end*

$I_{n_2} = n_1;$

*end*

*ii) Determine the general Chapman-Kolmogorov differential equation of the model based on the boundaries of the state space* $\Omega$ *and map the two dimensional form to one:*

*for* $n_2 = 0$ to $J_0$;

$N_{n_2} = N_{n_2-1} + I_{n_2} + 1;$

*end*

*for* $n_2 = 0$ to $J_0$

*for* $n_1 = 0 : I_{n_2}$

$\frac{dP(N_{n_2-1}+n_1+1)}{dt} = \triangle_1 + \triangle_2 + \triangle_3 + \triangle_4$

*where*

- *if* $(n_1-1, n_2) \in \Omega$ *then,* $\triangle_1 = \lambda_1 \alpha_1 (n_1-1, n_2) P(N_{n_2-1}+n_1) - n_1 \mu_1 P(N_{n_2-1} + n_1 + 1)$

*if* $(n_1 - 1, n_2) \notin \Omega$ *then,* $\triangle_1 = 0;$

- *if* $(n_1, n_2-1) \in \Omega$ *then,* $\triangle_2 = \lambda_2 \alpha_2 (n_1, n_2-1) P(N_{n_2-2}+n_1+1) - n_2 \mu_2 P(N_{n_2-1} + n_1 + 1)$

*if* $(n_1 - 1, n_2) \notin \Omega$ *then,* $\triangle_2 = 0;$

- *if* $(n_1+1, n_2) \in \Omega$ *then,* $\triangle_3 = (n_1+1) \mu_1 P(N_{n_2-1}+n_1+2) - \lambda_1 \alpha_1 (n_1, n_2) P(N_{n_2-1} + n_1 + 1)$

*if* $(n_1 + 1, n_2) \notin \Omega$ *then,* $\triangle_3 = 0;$

- *if* $(n_1, n_2+1) \in \Omega$ *then,* $\triangle_4 = (n_2+1) \mu_2 P(N_{n_2}+n_1+1) - \lambda_2 \alpha_2 (n_1, n_2) P(N_{n_2-1} + n_1 + 1)$

*if* $(n_1, n_2 + 1) \notin \Omega$ *then,* $\triangle_4 = 0;$

*end*

*end*

*iii) Implementing Lyapunov theorem and numerically integrate the differential equation model*

*Given the interval* $[t_0, t_f]$ *and step size* $N_{step}$;

$\triangle t = \frac{t_f - t_0}{N_{step}}$;

$t_i = t_0, t_e = t_i + \triangle t$;

*Set initial condition* $P(t_i) = P(0)$;

*for* $i = 1 : N_{step}$

- *Use the fifth order Runge-Kutta integration to solve* $P(t)$ *on* $t \in [t_i, te]$;
- *Calculate the desired blocking rate,* $\pi^d_{1,2}(t)$*, using (3.38). Solve numerically for the Capacity,* $C_1$ *and* $C_2$ *and update the link bandwidth such that* $C_{link} = C_1 + C_2$.
- *Update the time intervals:* $t_i = t_e, t_e = t_i + \triangle t$;

*end*

Note that, the blocking probabilities for both type of connection are given by:

$$\begin{aligned} \pi_1 &= \sum_{n_2}^{J_0} P(N_{n_2}) \\ \pi_2 &= \sum_{n_2=0}^{J_0} \sum_{n_1=I_{n_2+1}+1}^{I_{n_2}} P(N_{n_2-1} + n_1 + 1) \end{aligned} \tag{3.39}$$

An algorithm for three traffic classes, $k = 3$, is found at Appendix B.

# 4.0 SIMULATIONS AND DISCUSSION

In this chapter, we present numerical results illustrating the performance of the proposed Lyapunov based adaptive bandwidth control. We first consider the single traffic class case.

## 4.1 ADAPTIVE CAPACITY ALLOCATION FOR SINGLE TRAFFIC CLASS

### 4.1.1 Performance evaluation with different offered loads

We have implemented the developed capacity adjustment scheme in our fluid-flow modeling framework and have used a dynamically varying offered load to evaluate the behavior. We adopted the same scenarios as in [37] so that the results can directly be compared. For the purpose of this study, we have held the service rate constant at $\mu = 1$. We consider three different average offered loads: $\widehat{a} = 15, 45, 100$ to measure the effects that various loads have on the capacity adjustment scheme. The acceptable QoS call blocking rate is chosen to be $\pi^{desired} = 0.02$. Note that the blocking probability of 0.02 corresponds to 14.70 connections in the system for the average load of 15, using equation (3.6). The run time duration of each scheme was from $t_0 = 0$ to $t_f = 120$ where the first 10 time units were used for the start-up period. A sinusoidal function was used to represent the periodic, dynamic, offered load, $a(t) = \widehat{a} + (3\ sin(0.1(t + 20)))$. By keeping the service rate constant and modifying the average offered load, we have maintained the periodicity of the system for each case as well as the rate of change of the average load. Our interest is in observing the effect on the capacity adjustment scheme for different offered loads given that the rate and the periodicity of the load change was the same.

In Figures (23), (24), and (25) we plot the blocking rate versus time for the proposed adaptive capacity scheme with three different average load values respectively, $\widehat{a} = 15$, 45, and 100. As shown, the blocking rate is maintained within a range: $0.02 \pm 0.01$. Figures (26) and (27) show the capacity adjustment and number of connections in the system respectively for the three different load scenarios.

### 4.1.2 Performance evaluation with different desired blocking rates

In Figure (28), two different values for the desired blocking rate, 0.01 and 0.03 were considered. As shown in the figure, the curves of the blocking probability are following the given desired blocking rate which imply that the QoS requirement is satisfied. Figure (29) shows the corresponding capacity adjustments. One can see the smaller the desired blocking rate, the larger the capacity required.

### 4.1.3 Performance evaluation with different load characteristics

To evaluate the performance of the developed scheme to the speed of variations in the offered load, we vary the frequency of the load, specifically, we set $a(t) = 15 + 3\sin(\gamma(t + 20))$ and vary $\gamma = 0.1$, 0.5,and 0.05 . The offered load and the corresponding adjusted capacity for three frequencies cases are shown in Figures (30) and (31). As shown in Figures (30) and (31), the capacity curve is tracking the changes in the offered load. Note that the highest frequency case requires less capacity. Figures (32)-(34) show the blocking rates corresponding to the loads with different frequencies. As shown in the figures, the blocking probabilities are still maintained around the desired blocking rate, with the low frequency case having the poorest performance. Figure (35) shows the capacity adjustment behavior in response to a pulse change in the offered load

### 4.1.4 Comparison with other schemes in the literature

We compared our results with the ones obtained in [37] in which a scheme was built on the idea that if the connection blocking is beyond an acceptable level, or below a minimum

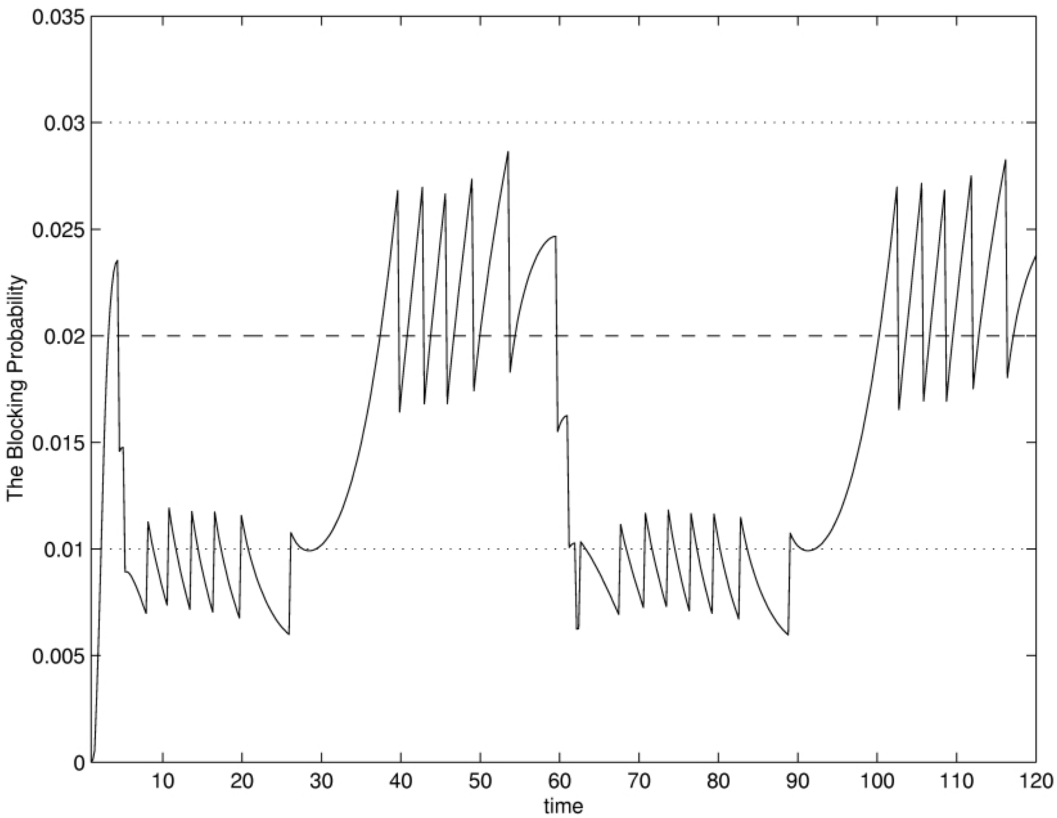


Figure 23: The blocking for the average offered load, $\widehat{a} = 15$

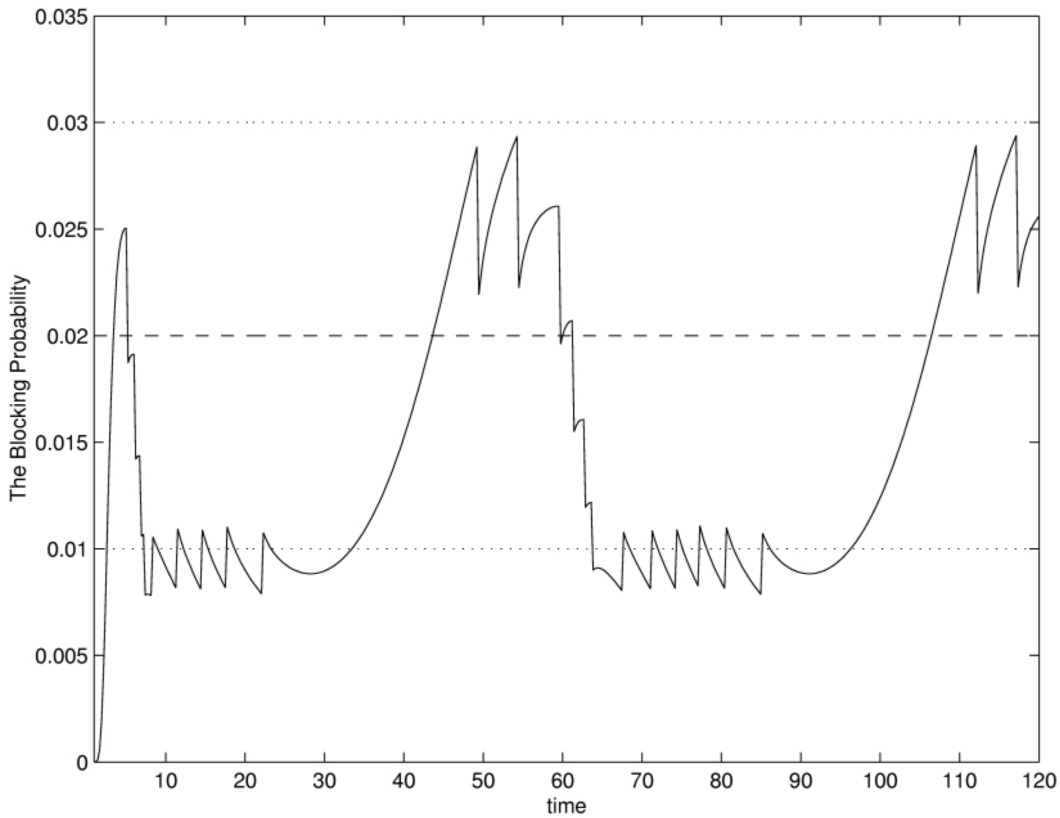


Figure 24: The Blocking for the average offered load, $\widehat{a} = 45$

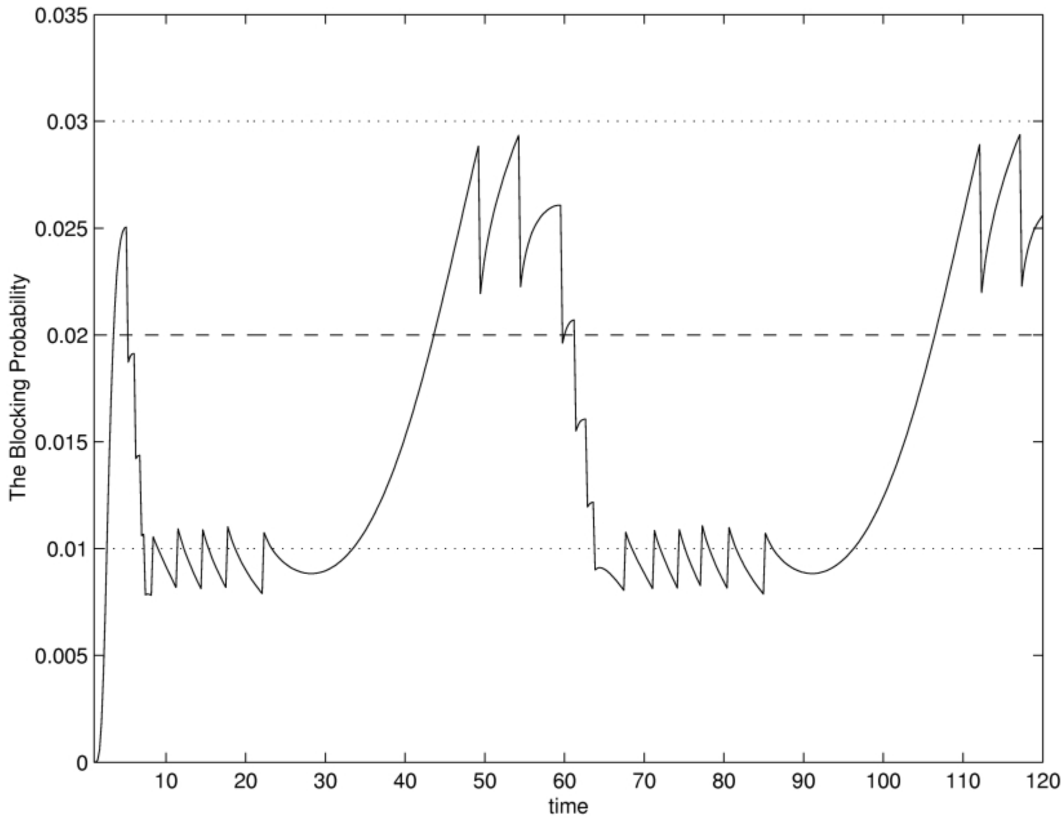


Figure 25: The Blocking for the average offered load, $\widehat{a} = 100$

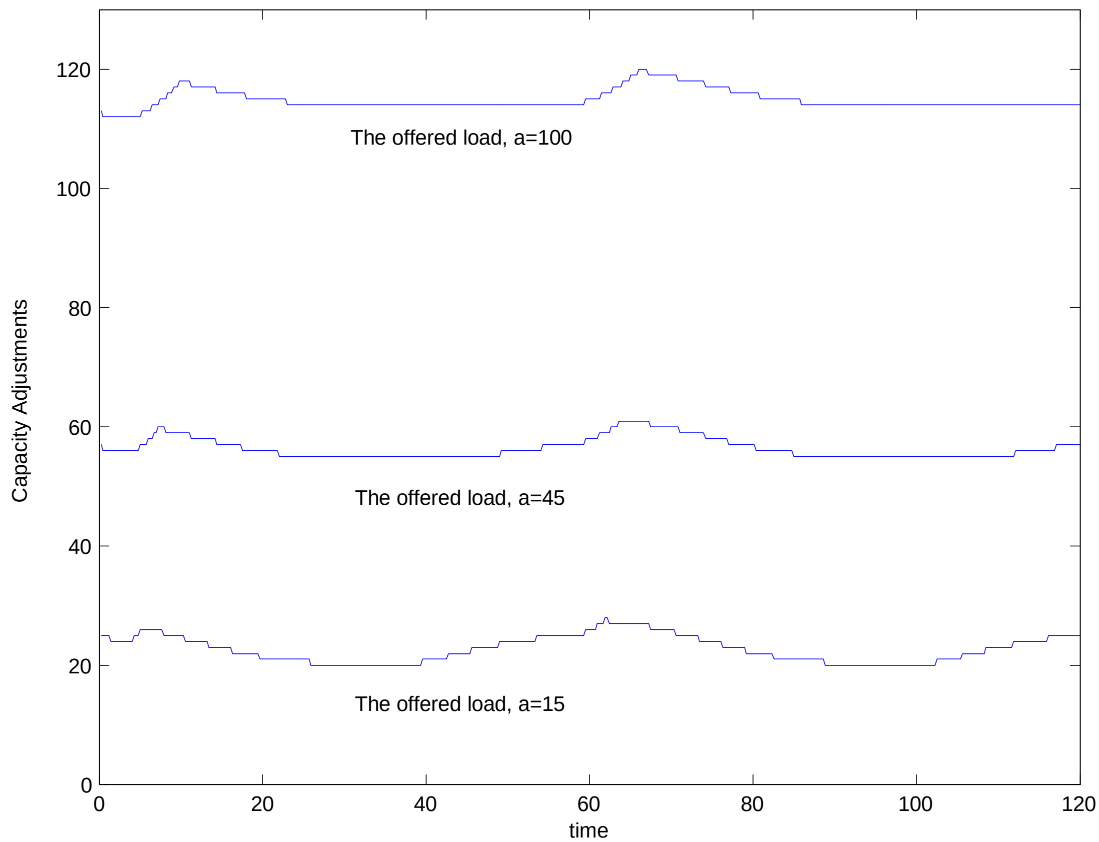


Figure 26: The Capacity Adjustments for the three offered Loads

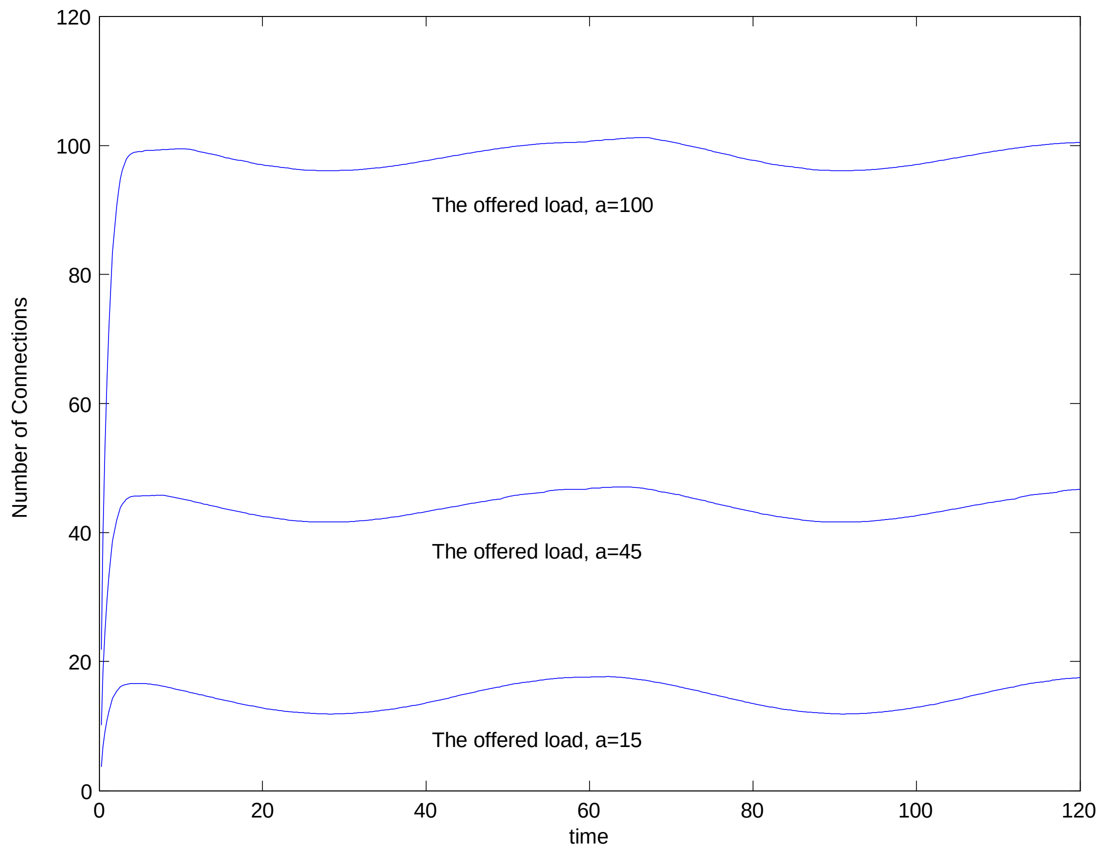


Figure 27: Number of Connections in the System for the Three Different Loads

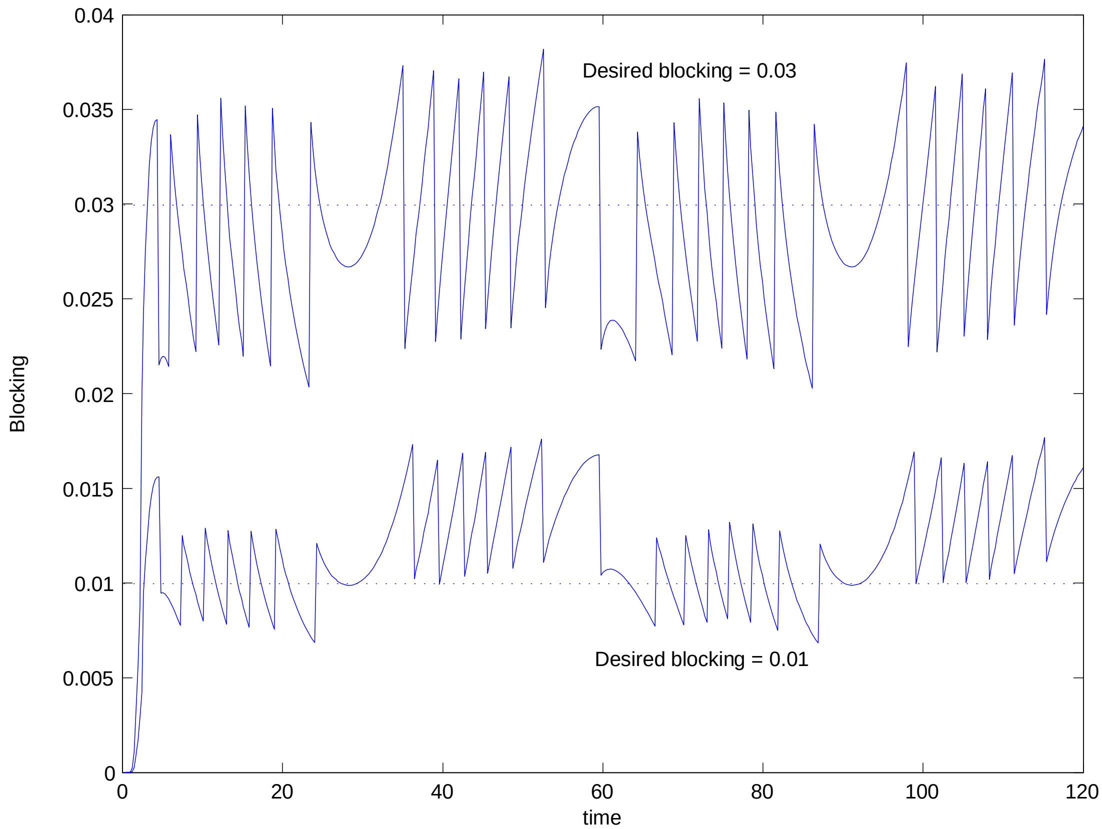


Figure 28: Blocking behavior with two different desired blocking rates and offered load of $a = 15 + 3\sin(0.1(t + 20))$

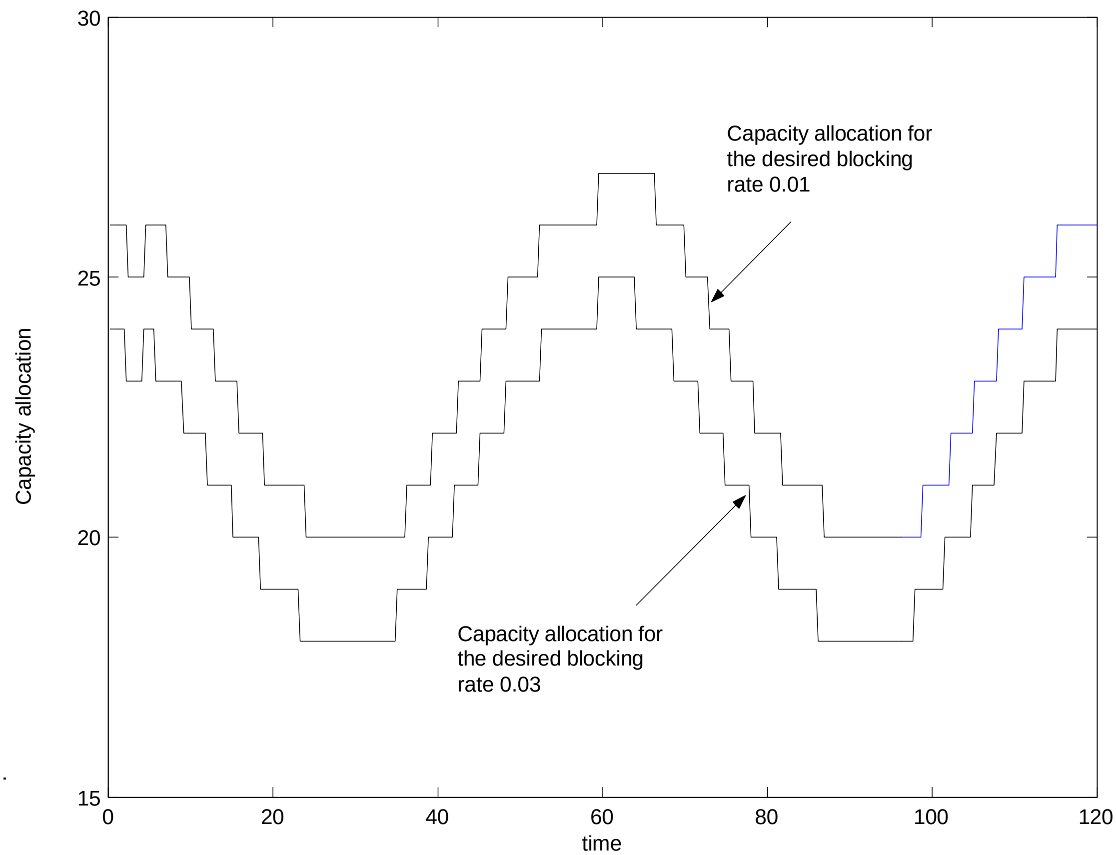


Figure 29: The capacity allocation with two different desired blocking rates and $a = 15 + 3sin(0.1(t + 20))$

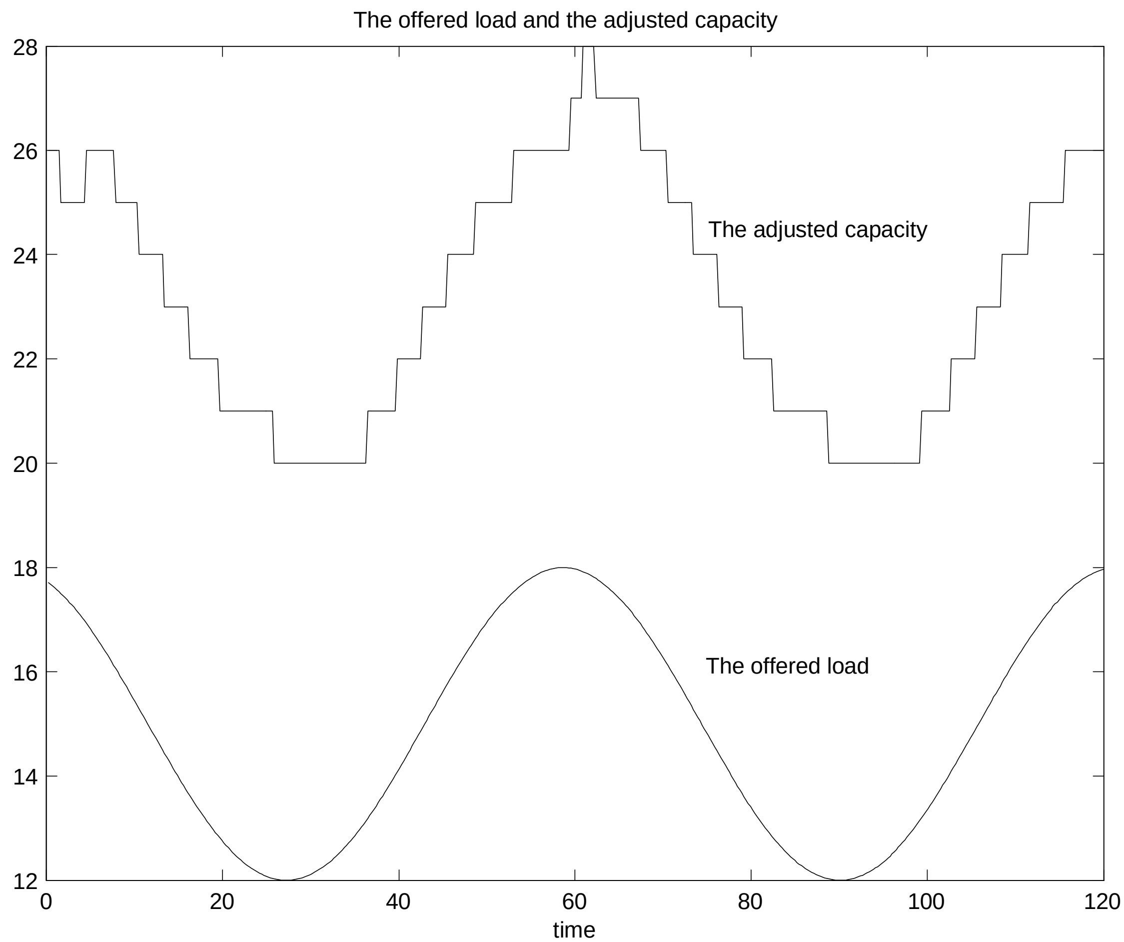


Figure 30: The offered load and the corresponding adjusted capacity with a load of medium-frequncy

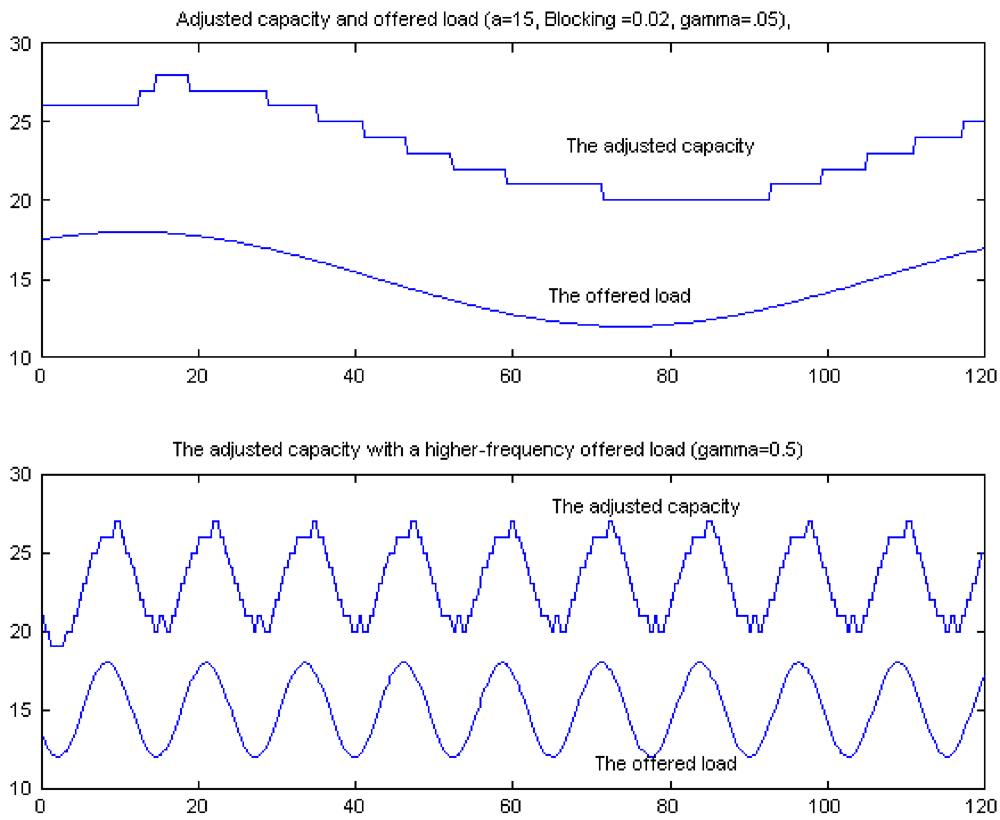


Figure 31: The offered load and the corresponding adjusted capacity with a load of low- and high-frequency

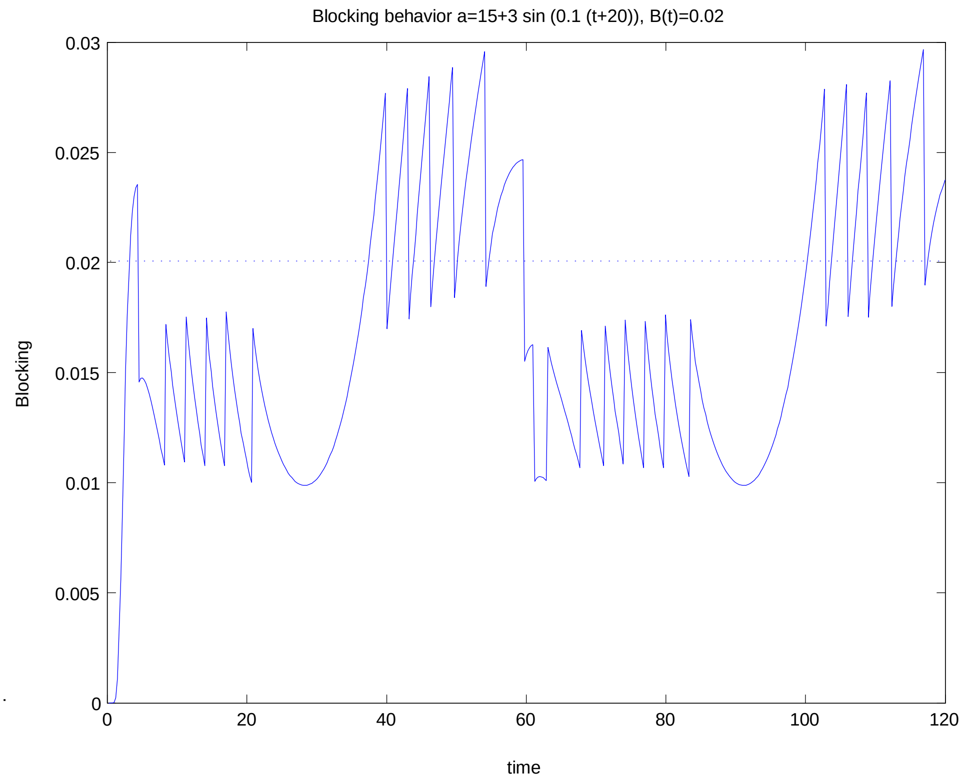


Figure 32: Blocking behavior with a medium-frequency offered load

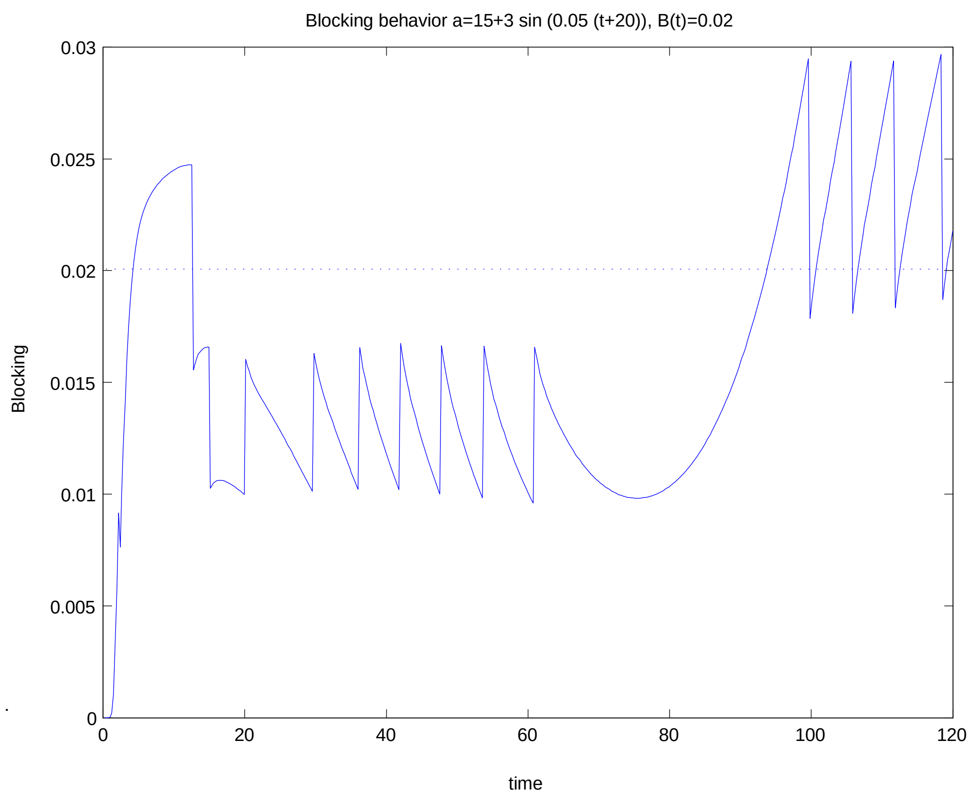


Figure 33: Blocking behavior with a low-frequency offered load

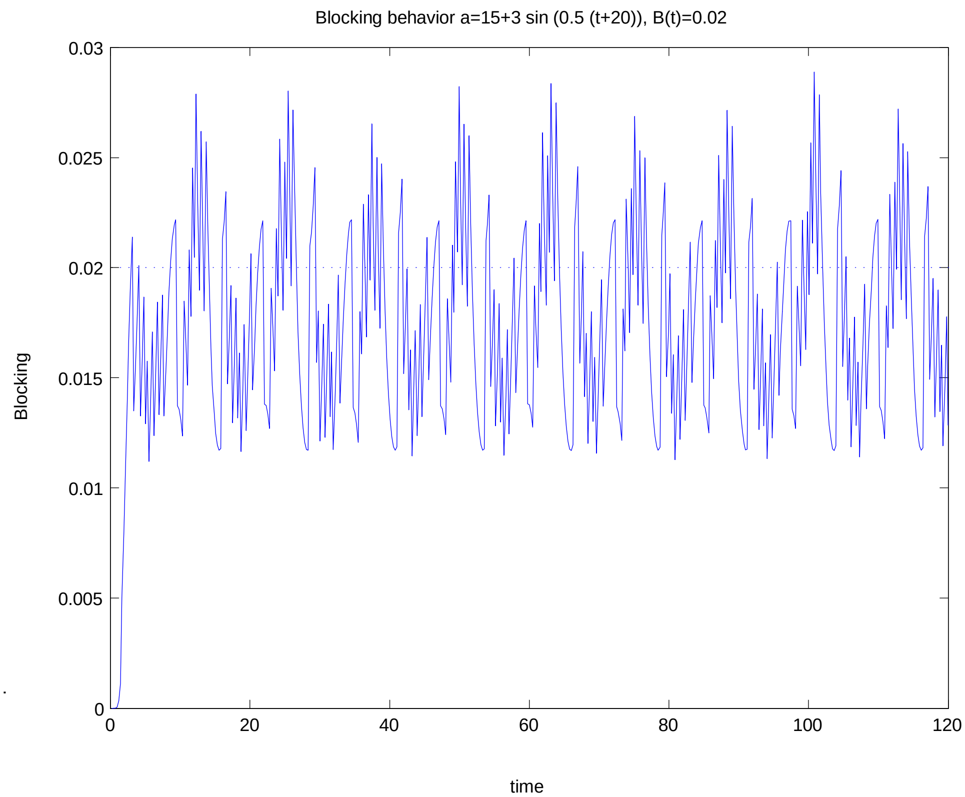


Figure 34: Blocking behavior with a high-frequency offered load

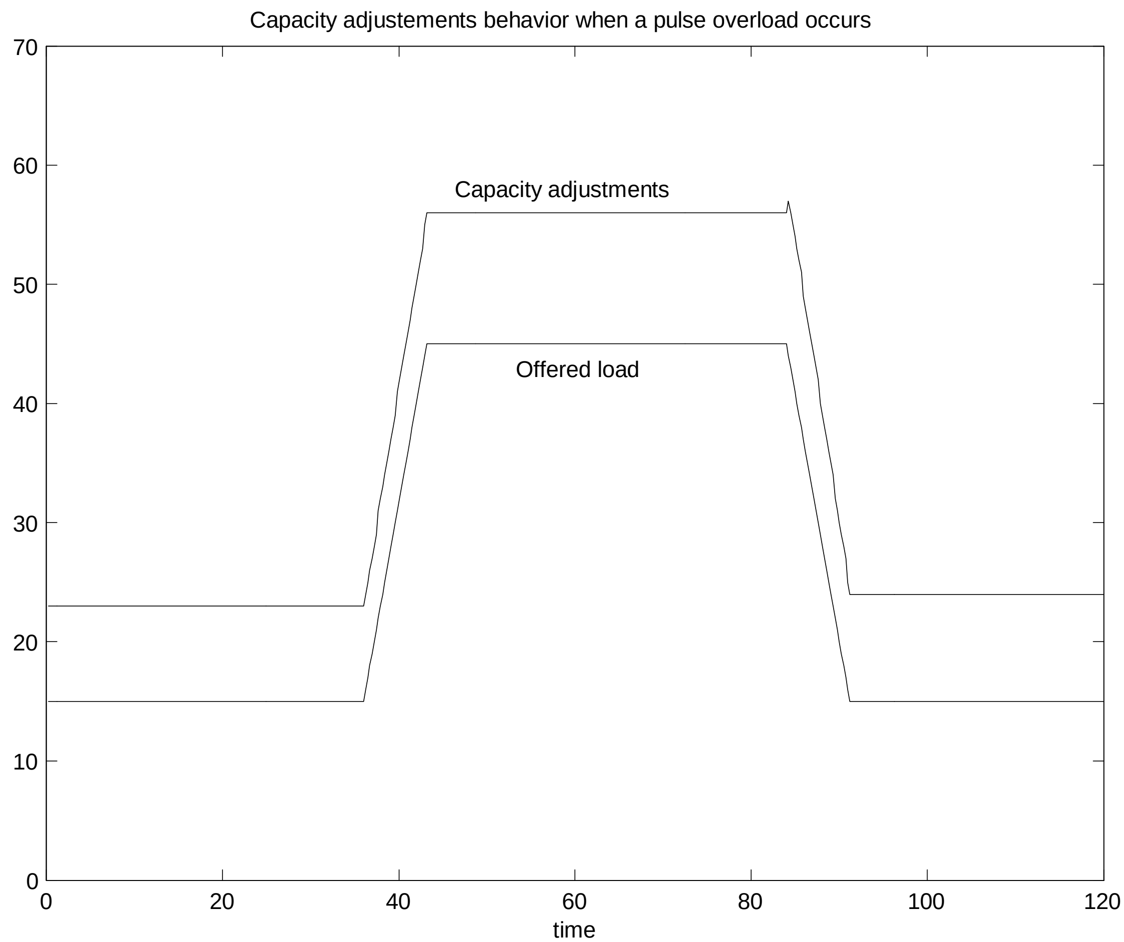


Figure 35: The offered load and the corresponding capacity for a load with a pulse change.

threshold, a request for additional capacity, or release of it, respectively, is desirable to provide an acceptable level of blocking. In that scheme proposed in [37], the actual amount to be adjusted is pre-set to a value, k, and the scheme can be summarized as follows. Let $b(t)$ be the blocking rate at time $t$, $b^{objective}$ denote the desired blocking rate and $b^{deviation}$ denote the allowable deviation from the desired blocking rate. The basic scheme in [37] is given as:

*If* $(b(t) < b^{objective}$ - $b^{deviation})$ *then*

$C(t) = C(t) - k$

*Else if* $(b(t) > b^{objective} + b^{deviation})$ *then*

$C(t) = C(t) + k$

*Else if* $(b^{objective}$ - $b^{deviation} \leq b(t) \leq b^{objective} + b^{deviation})$ *then*

*No adjustment*

*Endif*

For the purpose of comparison, we redeveloped that scheme and have plotted in Figure (36) the blocking versus time for $\widehat{a} = 15$ and $k = 1$, where the capacity adjustment luckily succeeded. Also, we show the case of $k = 3$, where the scheme failed in Figure (37). Clearly, our proposed scheme as shown Figure (23) outperforms the one in [37] as shown in comparison with Figure (37), since it does not require additional parameters, such as $k$, which might not be always available or easily determined. The proposed scheme has the advantage of that the capacity is automatically adjusted to meet the stability criteria.

### 4.1.5 Comparison with the static resource allocation

In this section, we compare the adaptive capacity allocation with static resource allocation to show the significance of the developed scheme. In both cases, the desired blocking rate is chosen to be 0.02 and the offered load of $a(t) = 15 + 3\ sin(0.1(t + 20))$. In the case of dynamic allocation the capacity is allocated as to keep the blocking probability around the desired rate, whilst in the static allocation scenario, the capacity is determined to keep the blocking probability below the desired blocking rate according to the peak load time. Figures (38) and (39) illustrate the two scenarios. From the figures, one can clearly see that static

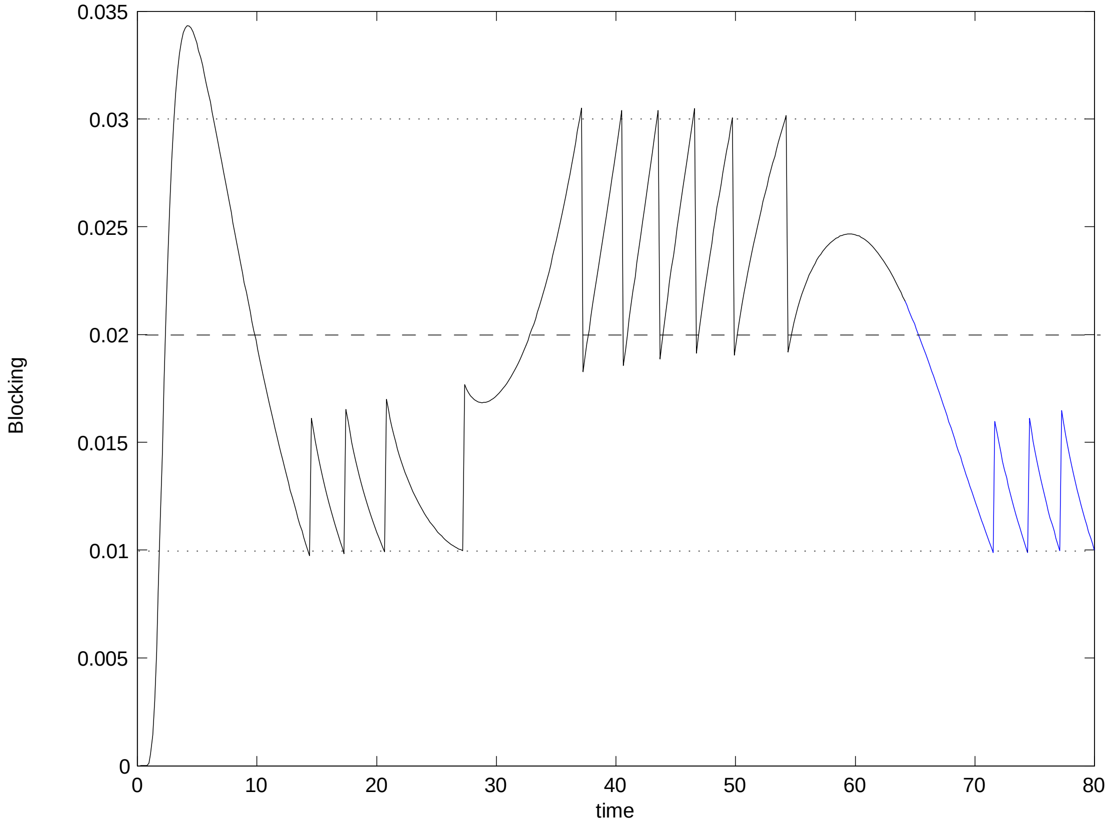


Figure 36: The blocking behavior of [37] scheme with k=1, and offered load of $a(t) = 15 + 3$ $sin(0.1(t + 20))$

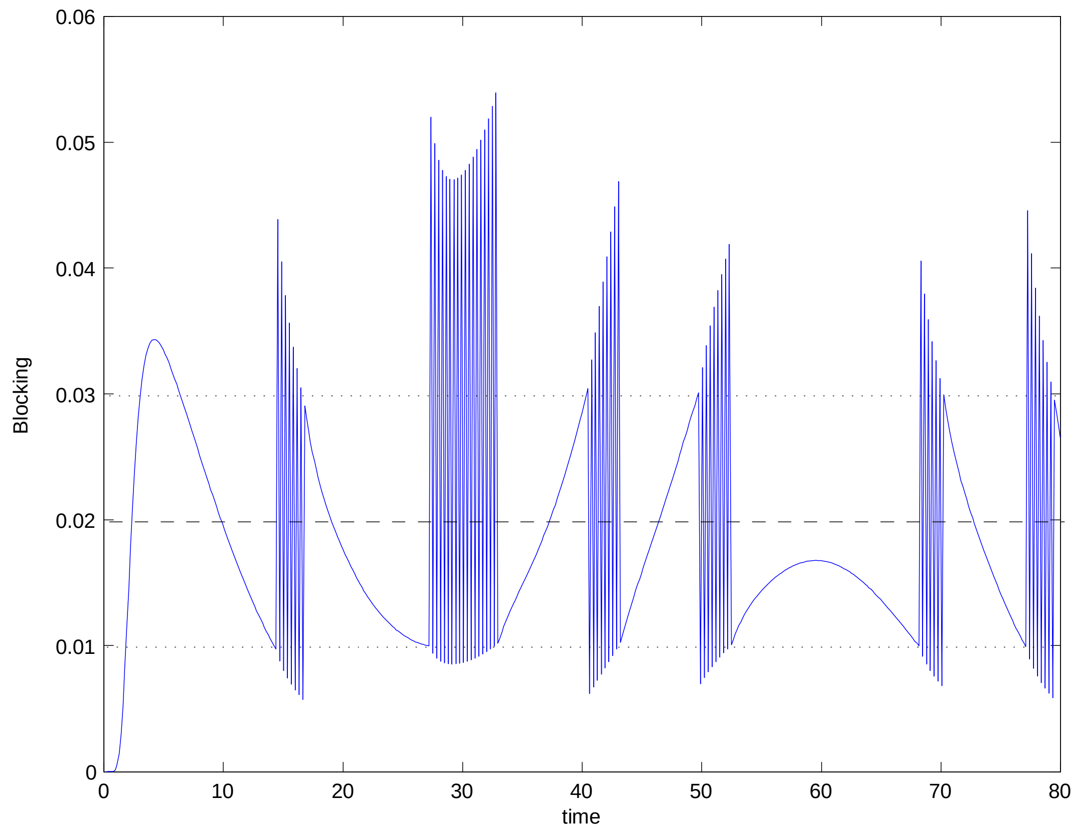


Figure 37: The blocking behavior of [37] scheme with k=3, and offered load of $a(t) = 15 + 3$ $sin(0.1(t + 20))$

allocation require much more bandwidth to be allocated and results in a overengineered system. Having shown the advantages of the Lyapunov scheme in the single traffic case, we now show the multiple traffic class case.

## 4.2 ADAPTIVE CAPACITY ALLOCATION FOR MULTIPLE CLASS TRAFFIC

### 4.2.1 Case 1: Multiple traffic classes with no coupling

Simulations with various parameters values for the offered load and desired blocking probabilities have been conducted as shown in Figures (40) through (44). Figure (40) illustrates the blocking behavior of two traffic classes with different desired blocking rates. In Figure (41), we plot the corresponding capacity adjustments curves, and the total capacity allocation, while Figure (42) shows the relationship between the offered load and the allocated capacity. The simulations have been extended for the case of ($k > 2$), specifically, for $k = 3$ since MPLS networks are typically concerned with three different traffic. The three offered loads in this simulation are: $a_1(t) = 15 + 3\ sin(0.1(t + 20))$, $a_2(t) = 15 + 3\ sin(0.2(t + 40))$, and $a(t) = 15 + 3\ sin(0.3(t + 10))$, and the desired blocking rates are $\pi_1^d = 0.04$, $\pi_2^d = 0.005$, and $\pi_3^d = 0.02$ respectively. Figures (43) and (44) show the blocking rate, the corresponding capacity adjustments, and the relation between the total offered load and capacity allocation.

### 4.2.2 Case 2: Multiple traffic classes with shared capacity

The fluid-flow modeling framework is used again to implement the adaptive capacity allocation. A sinusoidal function was used to represent the periodic, dynamic, offered load. Specifically, the offered traffic loads are: $a_1(t) = 1 + (0.5 \sin(0.2\ t))$ and $a_2(t) = 5 + (2\ sin(0.2(t + 20)))$ and the basic bandwidth units of each connection type are $m = 1$ and $m = 2$. Also, the service rates is held at constant value of $\mu_1 = 1$ and $\mu_2 = 1$. By keeping the service rate constant and modifying the average offered load, we have maintained the periodicity of the system for each case as well as the rate of change of the average load. The

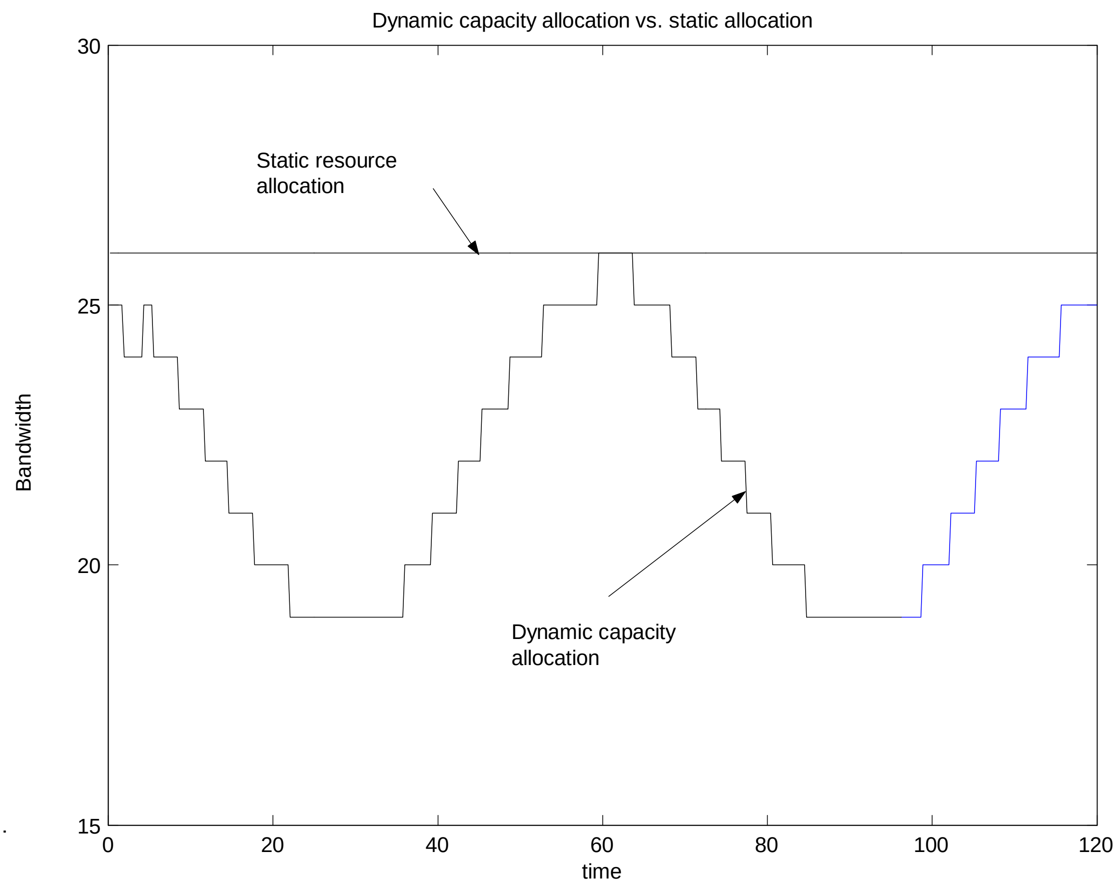


Figure 38: Dynamic capacity allocation versus static resource allocation

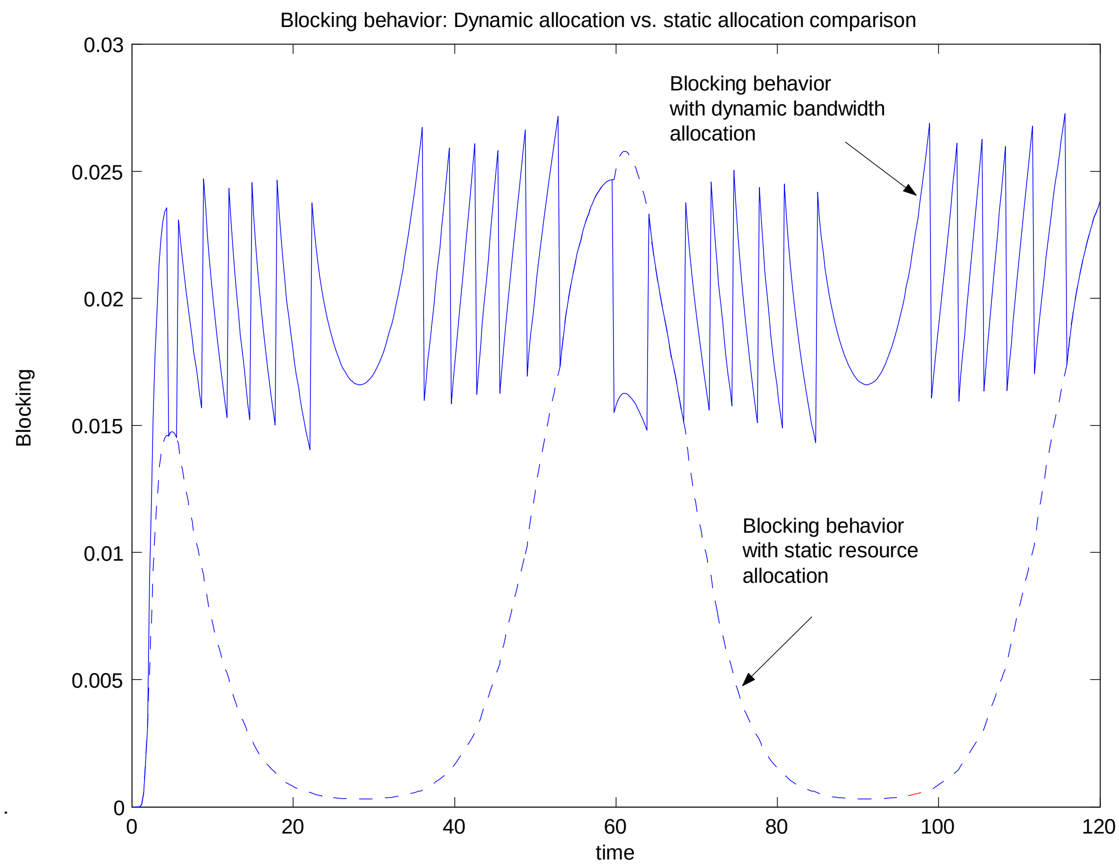


Figure 39: Blocking behavior with dynamic capacity allocation and static allocation

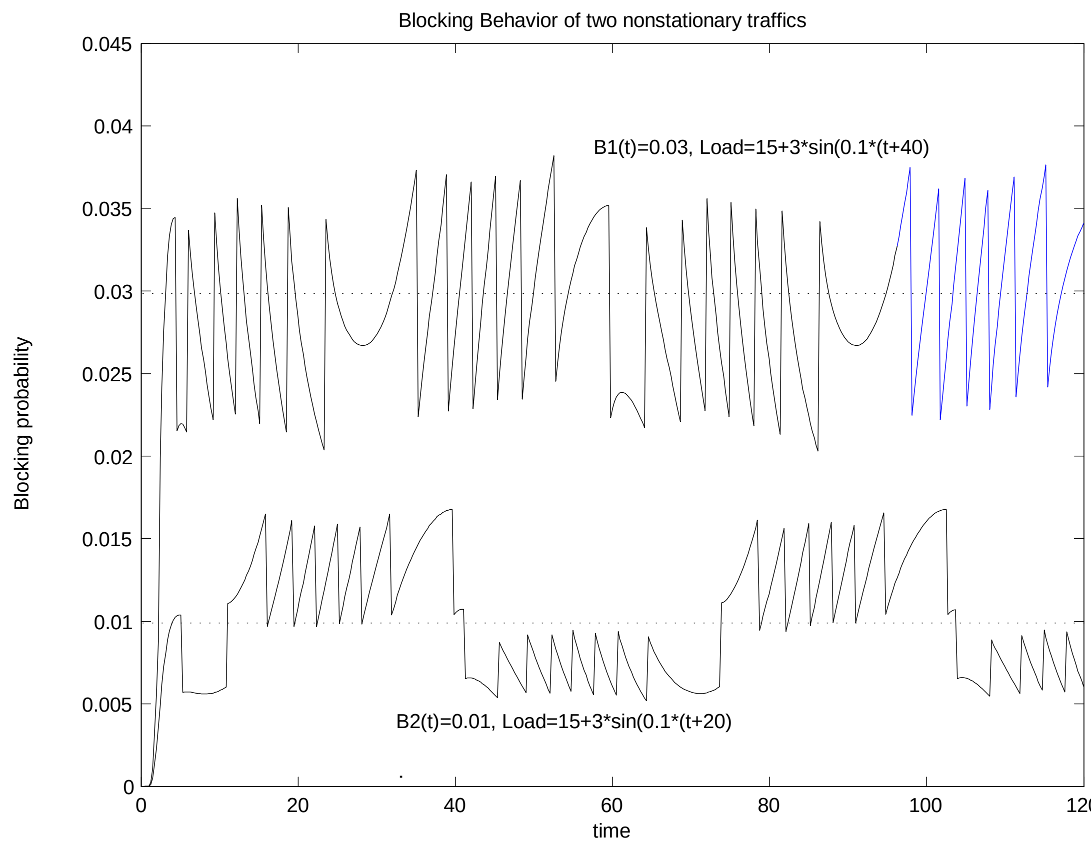


Figure 40: The blocking behavior of two traffic classes with different desired blocking rates (no coupling)

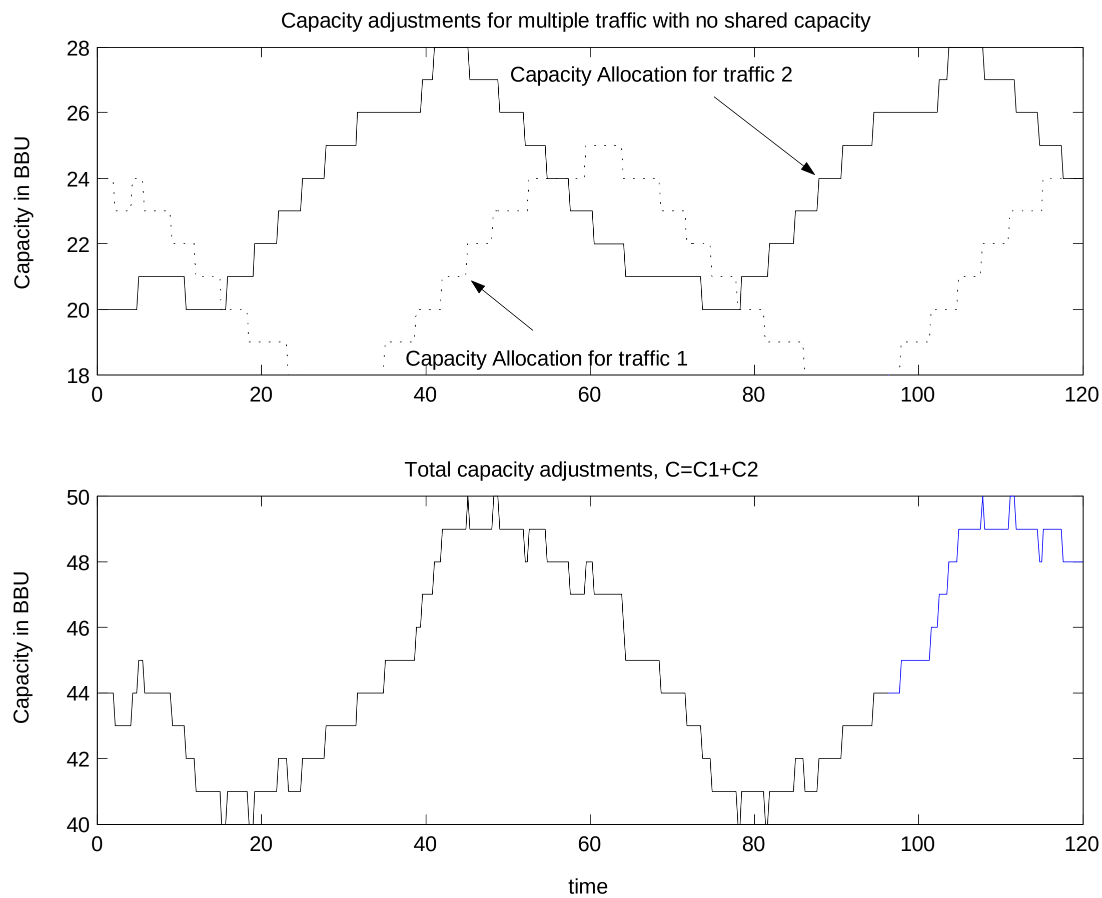


Figure 41: The capacity adjustments for two traffics with different desired blocking rates (no coupling)

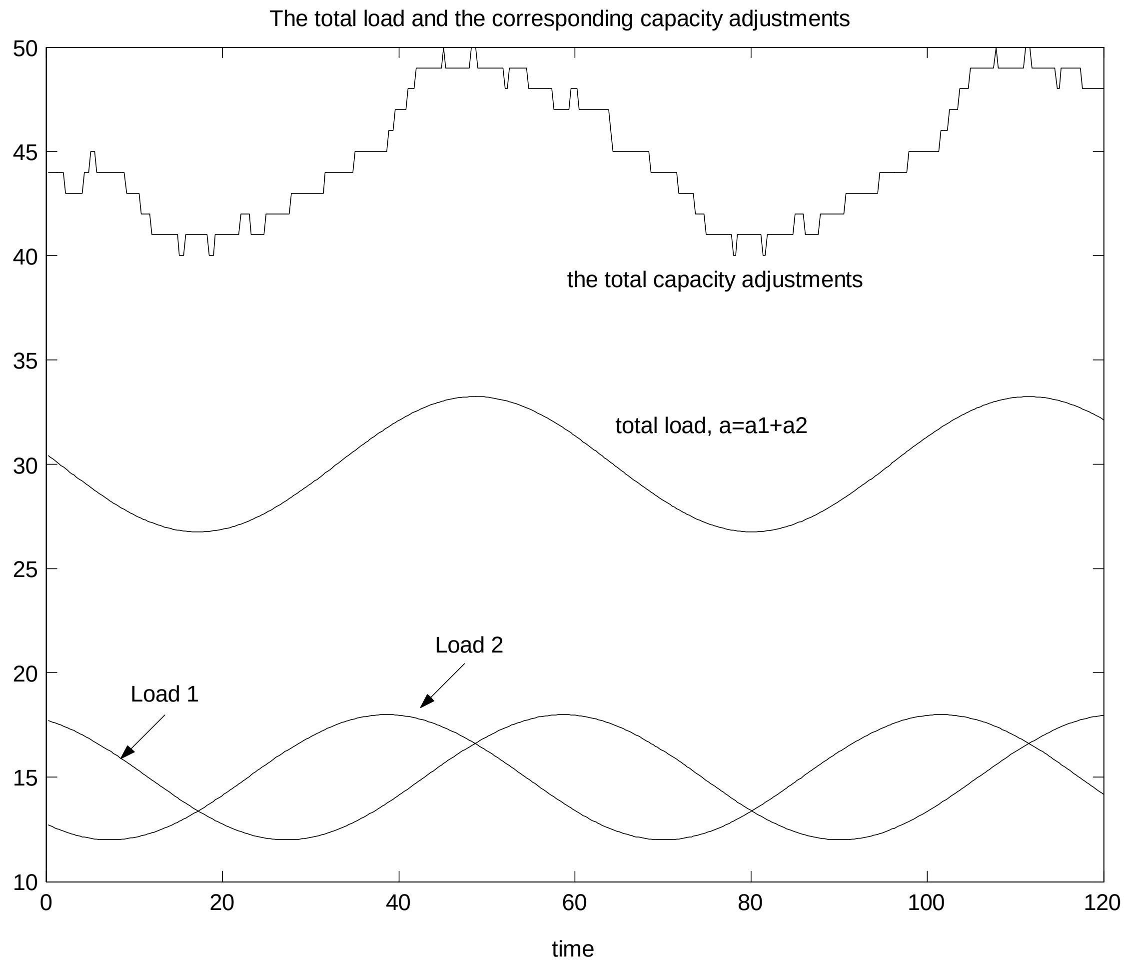


Figure 42: The total capacity adjustments and the offered load of the two traffics

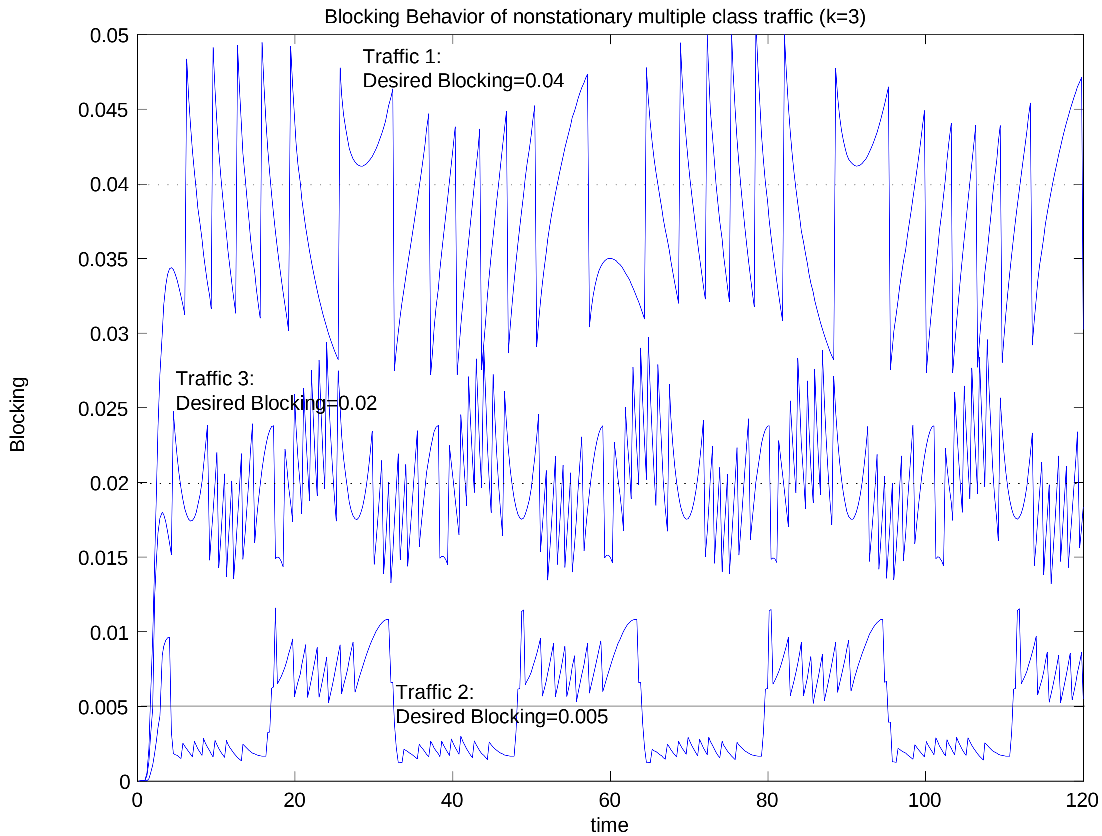


Figure 43: The blocking probability curves for k=3 case with three different desired blocking rates (no coupling)

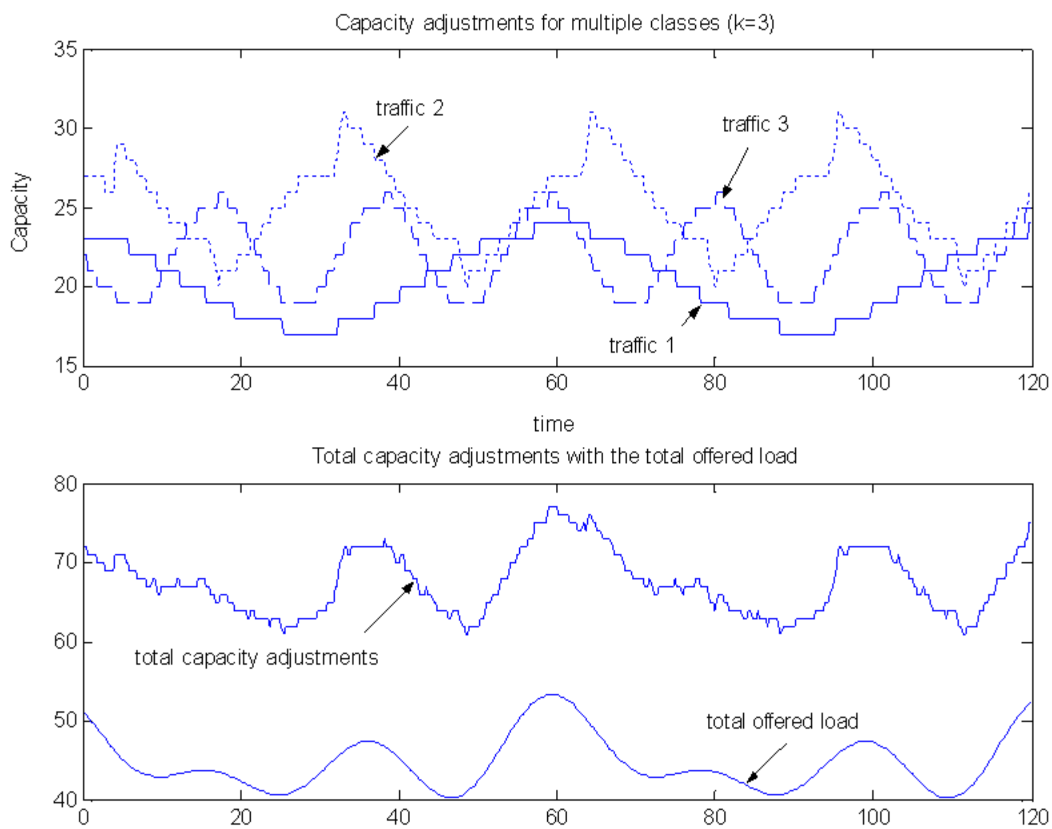


Figure 44: The resultant capacity allocation and total offered load (no coupling)

acceptable QoS call blocking rate is chosen to be $\pi_1^{desired} = 0.01$, and $\pi_2^{desired} = 0.02$ for the two incoming traffics. The run time duration of each scheme was from $t = 0$ to $t = 60$.

In this simulations, our interest is in observing the effect on the capacity adjustment scheme for the multiple offered loads. Figures (45) through (49) show the simulation outputs of our scheme. In this simulation, traffic class 2 has higher offered load and its blocking rate exceeds the desired blocking rate more frequently than class 1 as shown in Figure (45). Therefore, the capacity is more likely adjusted according to the offered load of traffic class 2. When the capacity is adjusted for class 2, the blocking rate of both classes drop, and the blocking rate of class 1 might be well below its desired blocking probability. Figure (46) show the capacity allocation of the two traffic classes. In Figure (47), we show how the adjusted capacity changes following the offered load the two traffic classes. Figures (48) and (49) illustrate the relationship between the offered load and the number of connections.

Figure (50) illustrates the case of three multiple traffic classes. In this scenario, traffic class 1 has a desired blocking rate of 0.004, $\hat{a} = 1$ and each traffic requires 4 basic bandwidth units, $m_1 = 4$. Traffic class 2 has desired blocking rate of 0.01, $\hat{a} = 5$ and each traffic requires 1 basic bandwidth unit, $m_2 = 2$. Traffic class 3 has desired blocking rate of 0.025, $\hat{a} = 5$ and each traffic requires 1 basic bandwidth unit, $m_3 = 1$.Figure (51) shows the corresponding adjusted capacity along with the total offered load. As shown in this scenario, traffic class 1 has a low desired blocking rate and high bandwidth requirement ($m_1 = 4$) for each traffic. Therefore, the blocking rate of this class (class 1) is more likely to go beyond its desired blocking probability, and hence, the capacity is more likely adjusted according to this traffic. As can be seen in Figure (50), the blocking probability of traffic class 1 is maintained around its desired blocking rate, while traffic class 3 has less blocking rate than its desired one because the bandwidth is completely shared among all traffics and the demand of the capacity increase requested by traffic class 1 to satisfy its requirement is higher than that of class 3.

In Figure (52), we compare the capacity allocation for the separate capacity case with the shared one for the same parameters, specifically, $a_1(t) = 1 + 0.5\ sin(0.2t)$, $a_2(t) = 5 + 2\ sin(0.2(t + 20))$, $\pi_1 = 0.01$, and $\pi_2 = 0.02$.In specific scenario, the separate capacity case has better performance and the bandwidth has more saving than the shared capacity since the two traffics used in this simulation has a great different characteristics, so that, the capacity allocated for a traffic class might be more than what it is needed by the other class that is sharing the same capacity.

## 4.3 LIMITATIONS

The scheme is designed to efficiently utilize the available bandwidth so that each traffic class receives the amount of capacity that gaurantees the required QoS. However, the scheme would be of no avail if the physical link bandwidth is limited and can not accomodate one or more offered traffic class. On the other hand, if the offered loads are almost stationary and use most of the link capacity, the developed scheme would not have a significant benefit since the capacity allocation is almost similar the static resource allocation schemes.

As mentioned earilier, the capacity is constrained by the relation of:

$$0 \leqslant \sum_{i=1}^{k} n_i(t)\ m_i \leqslant C, \quad \forall t \tag{4.1}$$

so if an incoming traffic with either expidited or assured service level requires bandwidth such as $m_i > C - \sum_{j=1}^{k} n_j m_j$, this traffic is blocked and cleared from the system.

## 4.4 NETWORK SERVICE MANAGEMENT

There are three traffic types in MPLS network, so the role of a network administrator would be to allocate a share of capacity based on the required QoS metric (i.e., the blocking probability). This would include grouping the traffic loads into different FECs so that each FEC is treated with different level of priority. Monitoring the network and hitorical reports would aid in impriving the service level agreements. Our develpoved scheme has the adavtage that the capacity is only utitized when it is need as shown in Figure (38). During the periods the capacity is surplus, the traffic with best efferot serive level is adequately admitted to the network. This way, the overall network performance is enhanced.

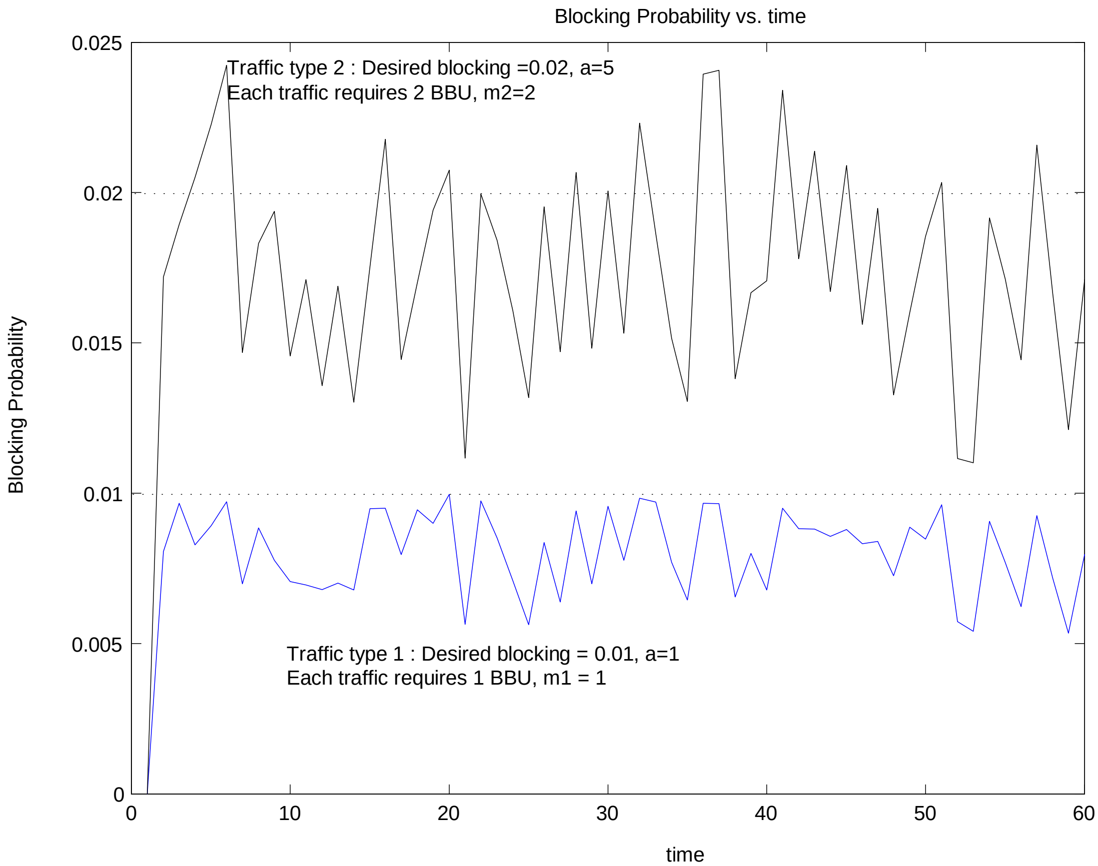


Figure 45: Blocking probabilty for multiple traffic classes with shared capacity

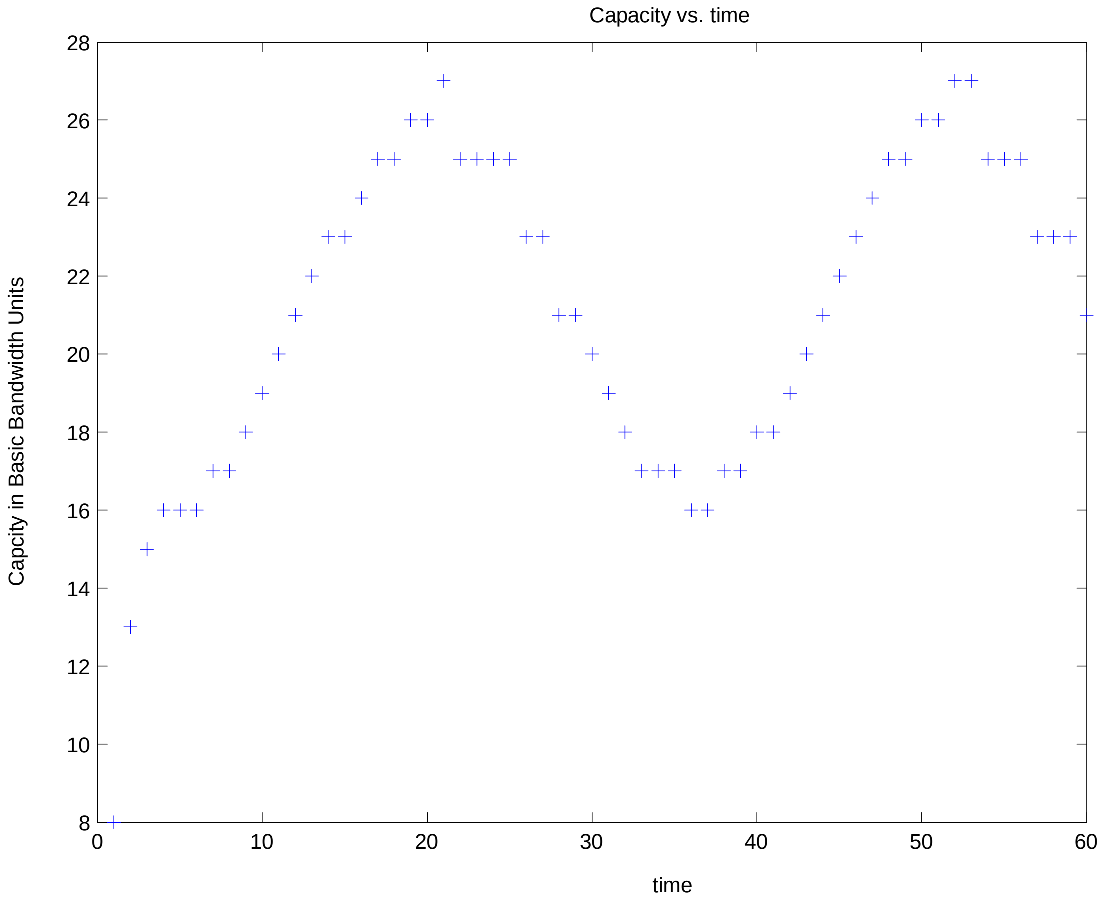


Figure 46: The capacity adjustment for mutiple traffic classes with shared capacity

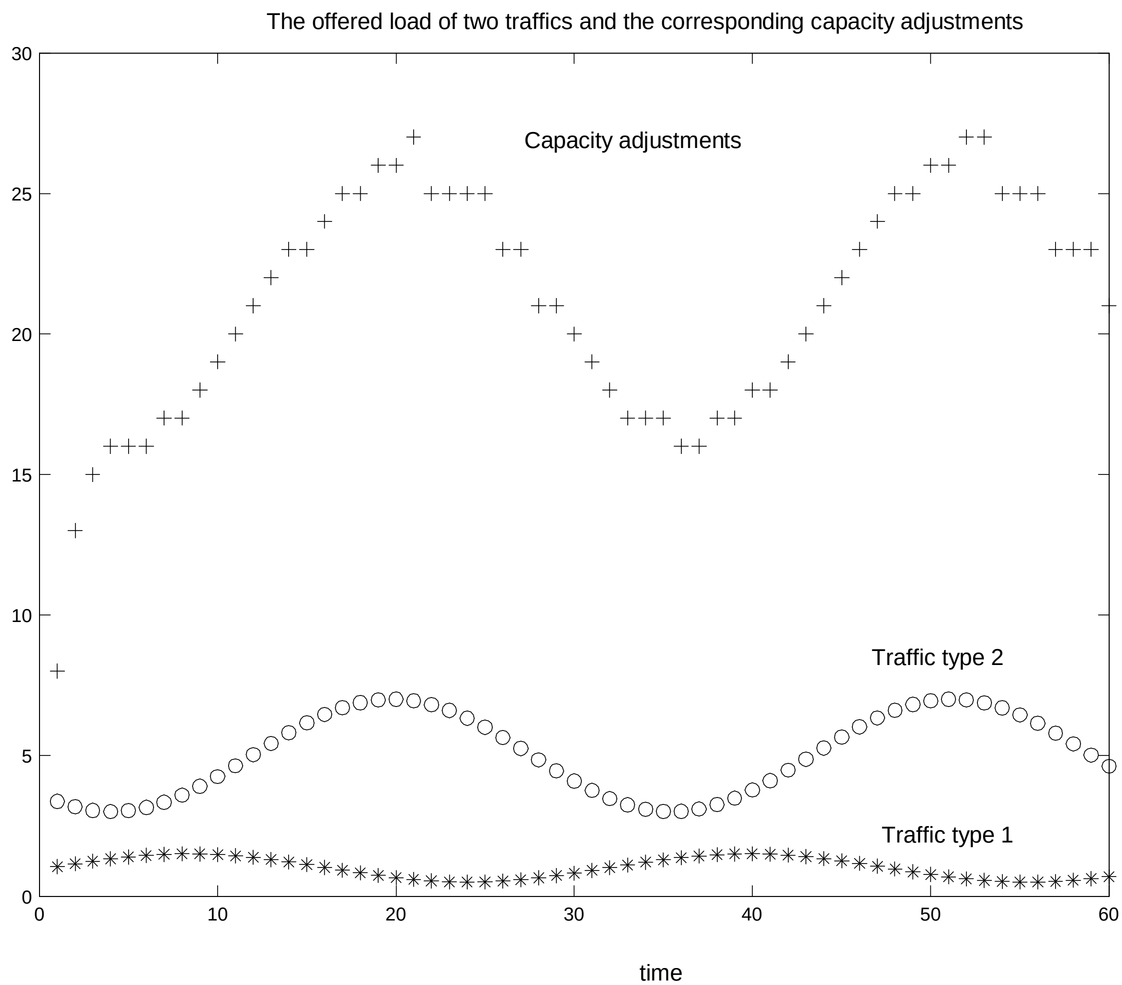


Figure 47: Capacity adjustments and the offered load of the multiple traffic classes of two types (shared capacity)

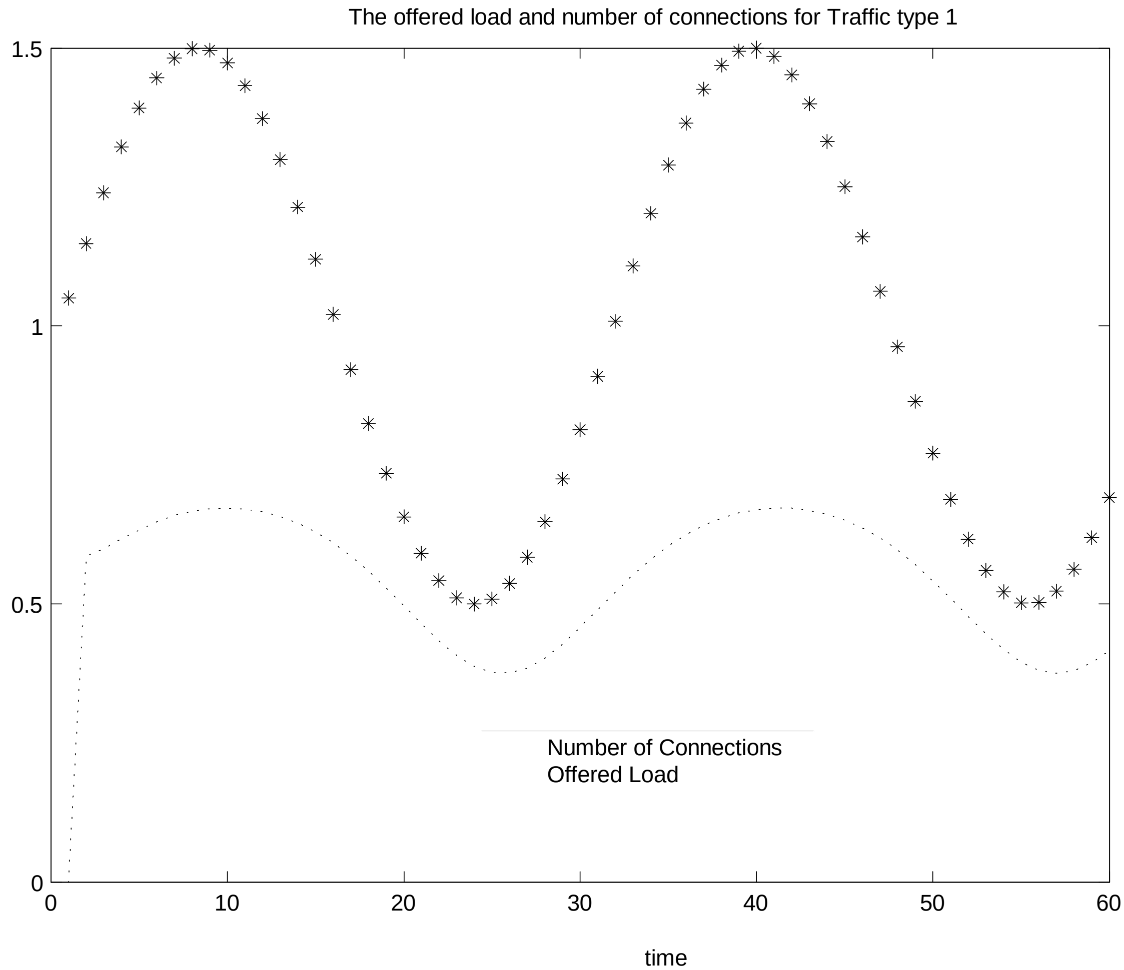


Figure 48: The offered load and the number of connections for traffic type 1 (multiple traffic classes case with shared capacity)

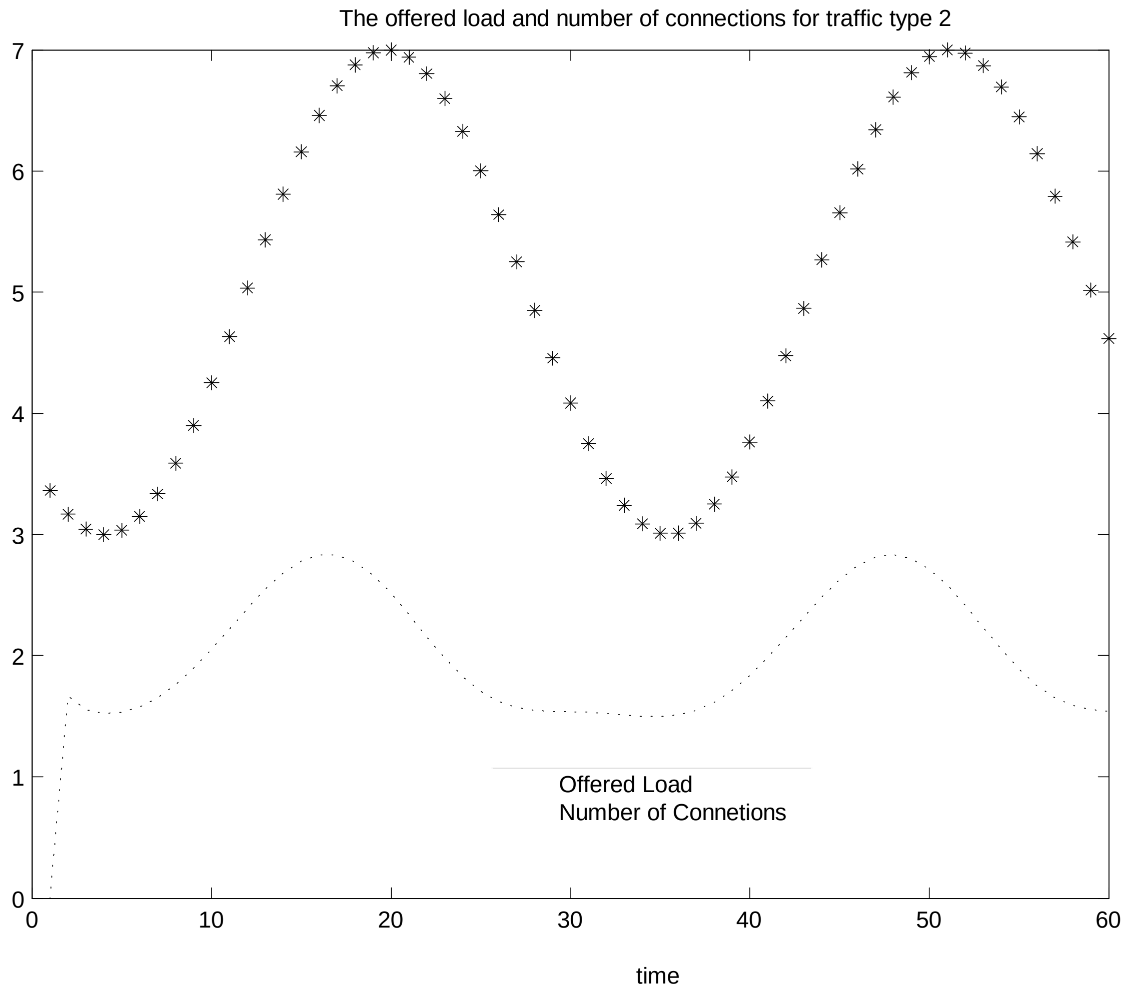


Figure 49: The offered load and the number of connections for traffic type 2 (multiple traffic classes case with shared capacity)

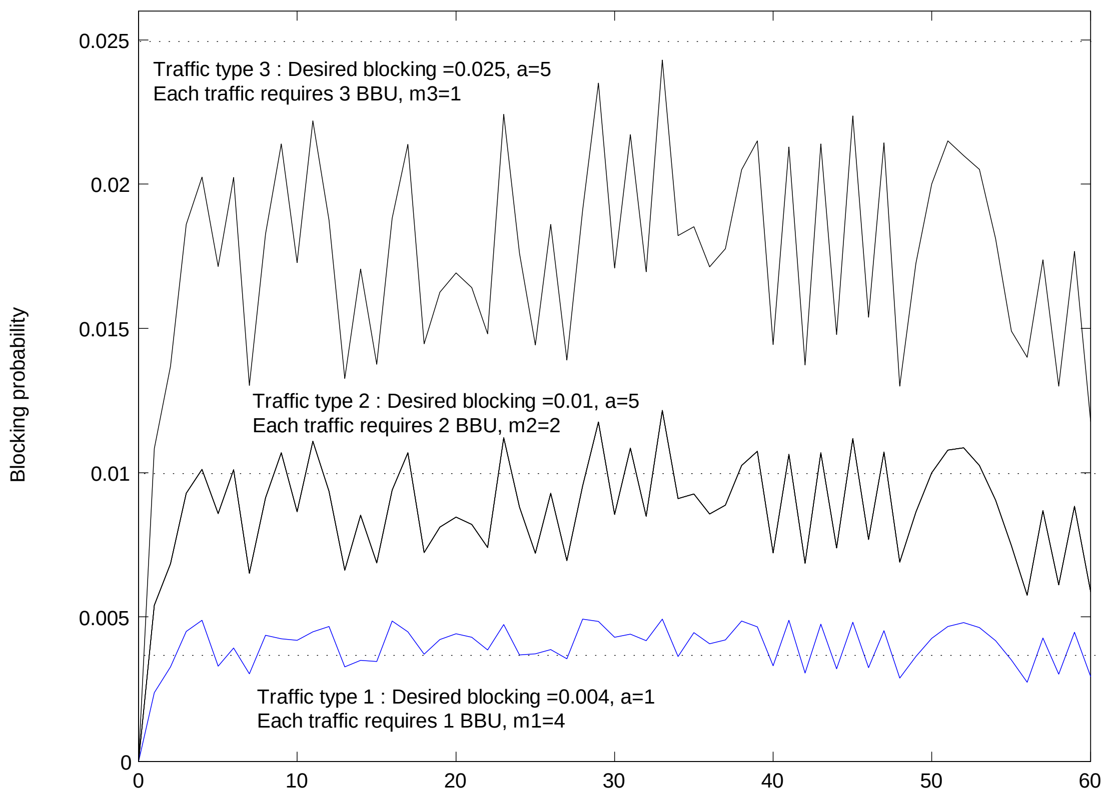


Figure 50: Blocking probabilities for three traffic classes with shared capacity

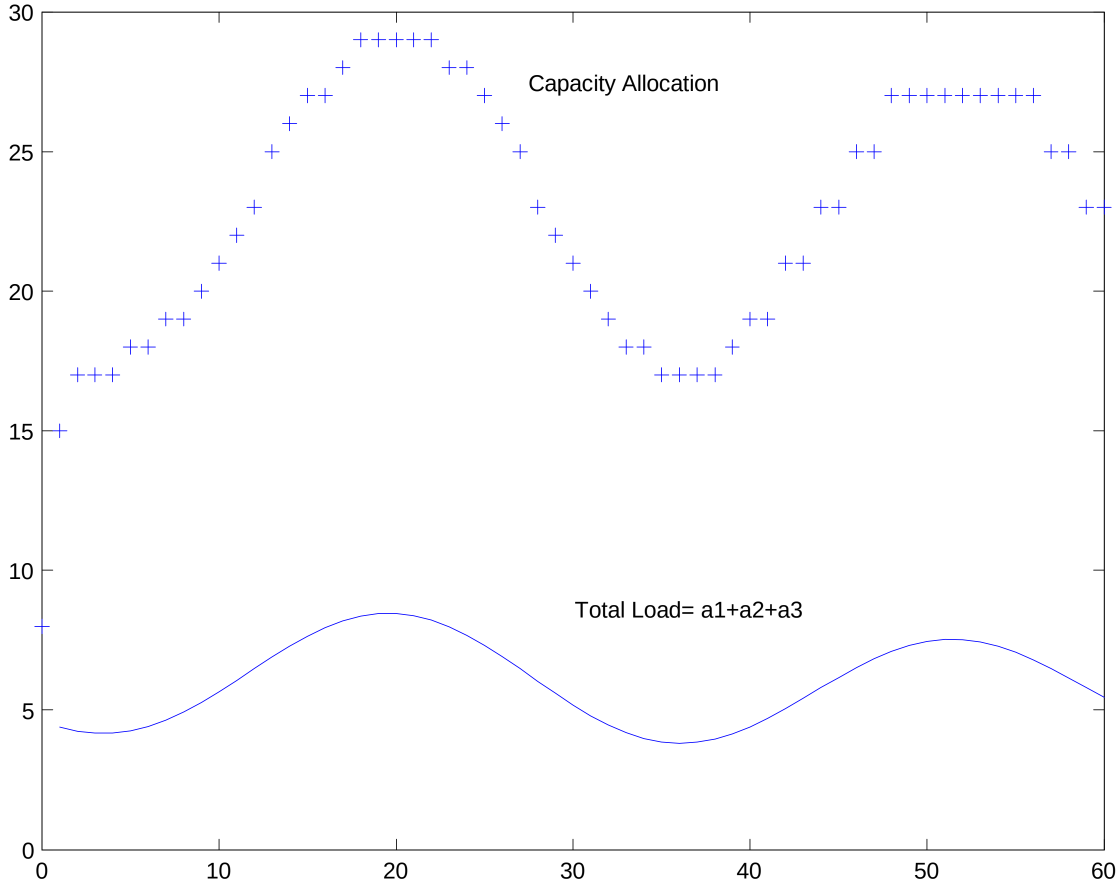


Figure 51: Capacity allocation for three traffic classes with shared capacity

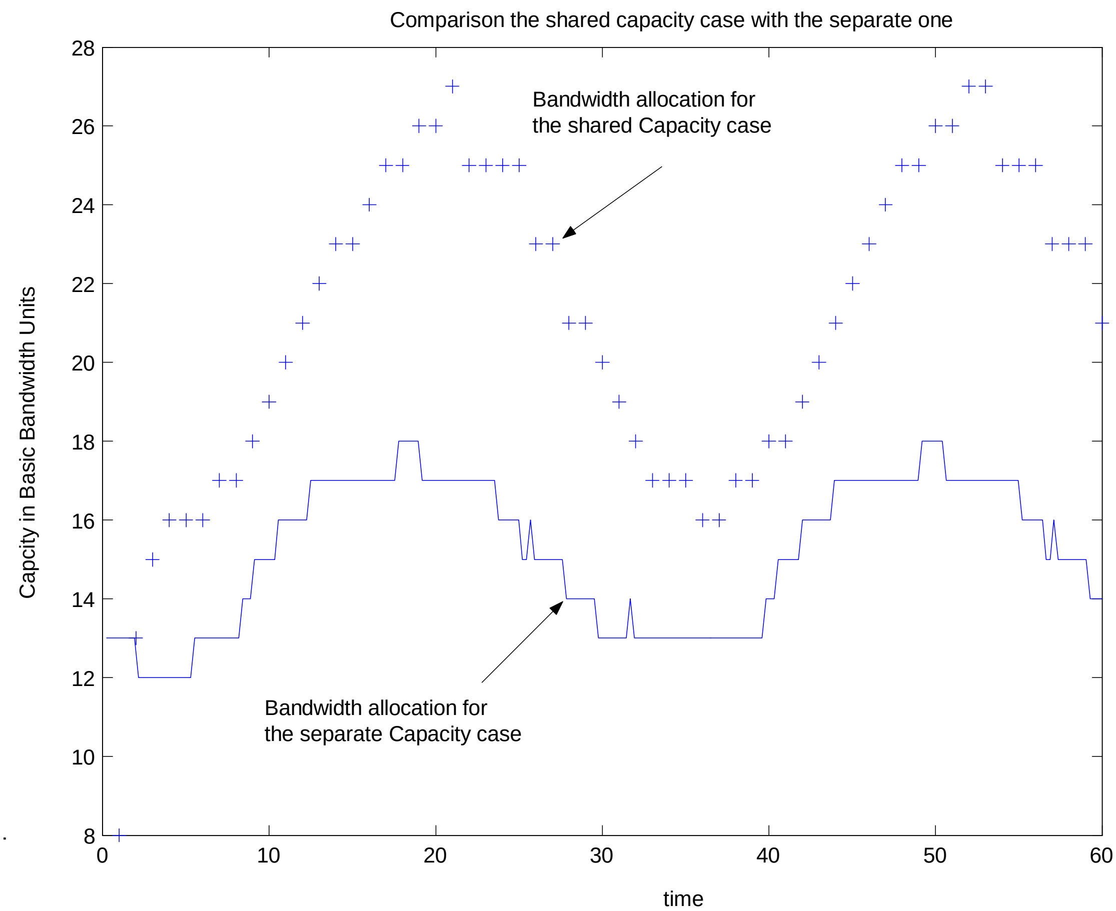


Figure 52: Comparison between the separate capacity case and the shared capacity case

# 5.0 CONCLUSION AND FUTURE WORK

## 5.1 SUMMARY

In this thesis, we developed a novel adaptive capacity allocation scheme for a dynamically reconfigurable network environment. The capacity allocation scheme was determined by the application of Lyapunov Stability Theory to a fluid flow model of network queuing behavior. The resulting Lyapunov based adaptive capacity control scheme seeks to maintain the connection blocking rate within an error bound around a desired QoS value.

Numerical results showing the effectiveness of the proposed scheme and its superiority over existing techniques were given. The work in this paper was first applied to a single traffic class system, then we extended it to the multiple traffic case.

## 5.2 CONTRIBUTION

In this thesis, our contribution can be summarized as follows:

- Study of congestion control in MPLS networks and identifying link bandwidth management as one of the open issues to investigate.
- Develop a framework for adaptive bandwidth control based on the application of the control theory to a fluid-flow model. The fluid-flow model provides a basis for different control strategies to be applied.
- A novel proposed adaptive capacity allocation scheme that works efficiently under the dynamic conditions for single traffic.
- An adaptive capacity allocation model for multiple traffic classes bandwidth allocation.

- A performance evaluation and comparative studies.

## 5.3 FUTURE WORK

- We will investigate the implementation of the developed scheme in real life using MPLS networks.
- The application of the proposed adaptive bandwidth control to wireless networks could be investigated.
- In this project, we used the standard assumption for arrival rate as Poisson distribution and service rate as exponential, so different processes may be assumed and investigated.
- Studying the integration of our scheme, which is applied at the connection level, with other schemes that are concerned with the adaptive capacity allocation at the packet level.

# APPENDIX A

## DERIVATION OF $F_{OUT}(T) = \mu X(T)$

For the M/M/c/c queueing model, the probability of blocking is given by the Erlang-B formula:

$\pi_C = \frac{\frac{a^C}{C!}}{\sum_{k=0}^{C} \frac{a^k}{k!}}$

The probability of no one in the system is:

$\pi_0 = \frac{1}{\sum_{k=0}^{C} \frac{a^k}{k!}}$

Also, the probability of $i$ customers in the system is:

$\pi_i = \frac{a^i}{i!}\pi_0$

Then, the flow out of the system is:

$f_{out} = \mu\pi_1(t) + 2\mu\pi_2(t) + ... + C(t)\mu\pi_{C(t)}$

Substituting for $\pi_i$ yields:

$f_{out} = \mu \left[a(t)\pi_0 + 2\frac{a(t)^2}{2!}\pi_0 + 3\frac{a(t)^3}{3!}\pi_0 + ... + C(t)\frac{a(t)^{C(t)}}{C(t)!}\pi_0\right]$

$f_{out} = \mu\ a(t)\ \pi_0 \left[1 + 2\frac{a(t)^{2-1}}{2!} + 3\frac{a(t)^{3-1}}{3!} + ... + C(t)\frac{a(t)^{C(t)-1}}{C(t)!}\right]$

$f_{out} = \mu\ a(t)\ \pi_0 \left[1 + \frac{a(t)^{2-1}}{(2-1)!} + \frac{a(t)^{3-1}}{(3-1)!} + ... + \frac{a(t)^{C(t)-1}}{(C-1)!}\right]$

$f_{out} = \mu\ a(t)\ \pi_0 \sum_{k=0}^{C(t)-1} \frac{a(t)^k}{k!}$

$f_{out} = \mu\ a(t)\ \pi_0 \left[\sum_{k=0}^{C(t)} \frac{a(t)^k}{k!} - \frac{a^{C(t)}}{C(t)!}\right]$

Substituting for $\pi_0$ yields:

$f_{out} = \mu\ a(t) \left[\frac{\sum_{k=0}^{C(t)} \frac{a(t)^k}{k!}}{\sum_{k=0}^{C(t)} \frac{a(t)^k}{k!}} - \frac{\frac{a(t)^{C(t)}}{C(t)!}}{\sum_{k=0}^{C(t)} \frac{a(t)^k}{k!}}\right]$

$$f_{out} = \mu\ a(t) \left[ 1 - \frac{\frac{a(t)^{C(t)}}{C(t)!}}{\sum_{k=0}^{C(t)} \frac{a(t)^k}{k!}} \right] = \mu a(t) \left[ 1 - \pi_{C(t)} \right] = \mu x(t).$$

## APPENDIX B

## ALGORITHM 3

The following algorithm is for a case of three traffic classes with shared capacity:

*i) Determine the boundaries of the two-dimensional matrix of* $\Omega$ *(i.e.,* $J_0$, $J_1$, $I_0$, $I_1$, $I_2$, ..., $I_{J_0}$*)*

$J_0 = \lfloor \frac{C}{m_2} \rfloor$;

$J_1 = \lfloor \frac{C}{m_3} \rfloor$;

$I_{n_2}(n_3, n_2) = \lfloor \frac{C - (\ n_2 m_2 \ + \ n_2 m_3)}{m_1} \rfloor$; $I_{n_2}$ *is a two-dimensional matrix (*$J_1 \times J_0$*) that determines the boundaries of the two-dimensional state transition.*

*where:* $n_2$*= 0,...,*$J_0$ *and* $n_3$*= 0,...,*$J_1$*.*

*ii) Determine the general Chapman-Kolmogorov differential equation of the model based on the boundaries of the state space* $\Omega$ *and map the two dimensional form to one:*

*for* $n_3 = 0$ to $J_1$;

  *for* $n_2 = 0$ to $J_0$;

    $N_{n_2}(n_3, n_2) = N_{n_2}(n_3, n_2 - 1) + I_{n_2}(n_3, n_2) + 1;$

      $//\ \ N_{n_2}\ is\ (J_1 \times J_0)\ matrix.$

      $//\ \ \ Where: N_{n_2}(n_3, n_2 - 1) = N(n_3 - 1,\ length(I_{n_2}(n_3, :));$

      $//\ \ \ N_{n_2}(-1, -1) = 0;$

  *end*

*end*

*for* $n_3 = 0$ to $J_1$

*for* $n_2 = 0$ to $J_0$

*for* $n_1 = 0 : I_{n_2}$

$\frac{dP(N_{n_2-1}+n_1+1)}{dt} = \triangle_1 + \triangle_2 + \triangle_3 + \triangle_4 + \triangle_5 + \triangle_6$

*where*

• *if* $(n_1 - 1, n_2, n_3) \in \Omega$ *then,* $\triangle_1 = \lambda_1\alpha_1(n_1 - 1, n_2, n_3)P(N(n_3, n_2 - 1) + n_1) - n_1\mu_1 P(N(n_3, n_2 - 1) + n_1 + 1)$

*if* $(n_1 - 1, n_2, n_3) \notin \Omega$ *then,* $\triangle_1 = 0;$

• *if* $(n_1, n_2 - 1, n_3) \in \Omega$ *then,* $\triangle_2 = \lambda_2\alpha_2(n_1, n_2 - 1, n_3)P(N(n_3, n_2 - 2) + n_1 + 1) - n_2\mu_2 P(N(n_3, n_2 - 1) + n_1 + 1)$

*if* $(n_1 - 1, n_2, n_3) \notin \Omega$ *then,* $\triangle_2 = 0;$

• *if* $(n_1 + 1, n_2, n_3) \in \Omega$ *then,* $\triangle_3 = (n_1 + 1)\mu_1 P(N(n_3, n_2 - 1) + n_1 + 2) - \lambda_1\alpha_1(n_1, n_2, n_3)P(N(n_3, n_2 - 1) + n_1 + 1)$

*if* $(n_1 + 1, n_2, n_3) \notin \Omega$ *then,* $\triangle_3 = 0;$

• *if* $(n_1, n_2 + 1, n_3) \in \Omega$ *then,* $\triangle_4 = (n_2 + 1)\mu_2 P(N(n_3, n_2) + n_1 + 1) - \lambda_2\alpha_2(n_1, n_2, n_3)P(N(n_3, n_2 - 1) + n_1 + 1)$

*if* $(n_1, n_2 + 1, n_3) \notin \Omega$ *then,* $\triangle_4 = 0;$

• *if* $(n_1, n_2, n_3 - 1) \in \Omega$ *then,* $\triangle_5 = \lambda_3\alpha_3(n_1, n_2, n_3 - 1)P(N(n_3 - 1, n_2 - 1) + n_1 + 1) - n_3\mu_3 P(N(n_3, n_2 - 1) + n_1 + 1)$

*if* $(n_1, n_2, n_3 - 1) \notin \Omega$ *then,* $\triangle_5 = 0;$

• *if* $(n_1, n_2, n_3 + 1) \in \Omega$ *then,* $\triangle_6 = (n_3 + 1)\mu_3 P(N(n_3 + 1, n_2 - 1) + n_1 + 1) - \lambda_3\alpha_3(n_1, n_2, n_3 + 1)P(N(n_3, n_2 - 1) + n_1 + 1)$

*if* $(n_1, n_2, n_3 + 1) \notin \Omega$ *then,* $\triangle_6 = 0;$

*end*

*end*

*end*

*iii) Implementing Lyapunov theorem and numerically integrate the differential equation model*

*Given the interval* $[t_0, t_f]$ *and step size* $N_{step}$;

$\triangle t = \frac{t_f - t_0}{N_{step}}$;

*$t_i = t_0, t_e = t_i + \triangle t$;*

*Set initial condition $P(t_i) = P(0)$;*

*for $i = 1 : N_{step}$*

- *Use the fifth order Runge-Kutta integration to solve $P(t)$ on $t \in [t_i, te]$;*
- *Calculate the desired blocking rate, $B^d_{1,2,3}(t)$, using (B.1). Solve numerically for the Capacity, $C_1$ and $C_2$ and update the link bandwidth such that $C_{link} = C_1 + C_2 + C_3$.*
- *Update the time intervals: $t_i = t_e, t_e = t_i + \triangle t$;*

*end*

$$B^d_k(t) \leq -\frac{\mu_k}{\lambda_k} x_k - \frac{\dot{x}_d}{\lambda_k} + 1 \tag{B.1}$$

Note that, the blocking probabilities for both type of connection are given by:

$$\begin{aligned} &B_i = 1 - P(\textit{type } i \textit{ connection is accepted}) \\ &Bi = 1 - \textstyle\sum_{(n_1,n_2,n_3)\in\Omega} \alpha_i(n_1, n_2, n_3)\ P(t; n_1, n_2, n_3) \end{aligned} \tag{B.2}$$

# APPENDIX C

## ACRONYMS

| | |
|---|---|
| ABC | Adaptive Bandwidth Control |
| ATM | Asynchronous Transfer Mode |
| BBU | Basic Bandwidth Unit |
| BGP | Border Gateway Protocol |
| CoS | Class of Service |
| CR–LDP | Constraint-based Routing-Label Distribution Protocol |
| DiffServ | Differentiated Services |
| DLCI | Data Link Connection Identifier |
| ER | Explicit Routing |
| FEC | Forward Equivalence Class |
| FTP | File Transfer Protocol |
| IETF | The Internet Engineering Task Force |
| IntServ | Integrated Services |
| IP | Internet Protocol |
| LAN | Local Area Network |
| LDP | Label Distribution Protocol |
| LER | Label Edge Router |
| LIB | Label Information Base |
| LSP | Label-Switched Path |
| LSR | Label Switching Router |

| | |
|---|---|
| MAC | Media Access Control |
| M/M/c/c | A queueing model with Poisson interarrival, exponential service distribution, c customers, and c servers |
| MPLS | Multi-Protocol Label Switching |
| OSPF | Open Shortest Path First |
| PHY | Physical Layer |
| PIM | Protocol-Independent Multicast |
| PNNI | Private Network-to-Network Interface |
| PPP | Point to Point Protocol |
| PSFFA | Pointwise Stationary Fluid Flow Approximation |
| QoS | Quality of Service |
| RSVP | Resource Reservation Protocol |
| SLA | Service Level Agreement |
| SONET | Synchronous Optical Network |
| TCP | Transmission Control Protocol |
| TTL | Time To Live |
| UDP | User Datagram Protocol |
| VC | Virtual Channel |
| VP | Virtual Path |
| VPI/VCI | virtual path identifier/virtual channel identifier |
| VPN | Virtual Private Network |

## BIBLIOGRAPHY


[1] White Paper, "Multiprotocol Label Switching (MPLS)", http://www.iec.org/online/tutorials/mpls, sponsored by Intel.

[2] Daniel O. Awduche, Bijan Jabbari, "Internet traffic engineering using multi-ptotocol label switching (MPLS)", Computer Networks 40 (2002) 111–129.

[3] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol label switching (mpls) architecture," IETF RFC 3031, January 2001.

[4] B. Davie, Y. Rekhter, "MPLS: Technology and Applications," Morgan Kaufmann Publisher, 1st edition, 2000.

[5] R. C. Palazzo, "A Lyapunov Approach to Dynamic Bandwidth Allocation in B-ISDN," M.S. thesis, School of Electrical Engineering, Clemson University, 1994.

[6] A. Pitsillides, P. Ioannou, and L. Rossides, "Congestion Control for Differentiated Services using Non-linear Control Theory," IEEE 6th Proc. Symp. Comp. and Commun., 2001, pp. 726–33.

[7] S. Rampal et al., "Dynamic Resource Allocation Based on Measured QoS," Technical Report TR 96-2, Center for Advanced Computing and Commun., North Carolina State University, Jan. 1996.

[8] I. Hsu and J. Walrand, "Dynamic Bandwidth Allocation for ATM Switches," J. Applied Probability, vol. 33, no. 3, 1996, pp. 758–71.

[9] R. R.-F Liao and A. T. Campbell, "Dynamic Core Provisioning for Quantitative Differentiated Service," Proc. IEEE/IFIP IWQoS'01, Karlsruhe, Germany, June 2001.

[10] P. Siripongwutikorn, S. Banerjee, and D. Tipper, "Adaptive Bandwidth Control for Efficient Aggregate QoS Provisioning," Proc. IEEE GLOBECOM'02, Taipei, Taiwan, Nov. 2002.

[11] G. Kesidis, "Bandwidth Adjustments Using Online Packet-level Measurements," SPIE Conf. Performance and Control of Network Systems, Boston, MA, Sept. 1999.

[12] N. Christin, J. Liebeherr, and T. F. Abdelzaher, "A Quantitative Assured Forwarding Service," Proc. IEEE INFOCOM'02, New York, NY, June 2002.

[13] S. Chong, S.-Q. Li, and J. Ghosh, "Predictive Dynamic Bandwidth Allocation for Efficient Transport of Real-Time VBR Video over ATM," IEEE JSAC, vol. 13, no. 1, pp. 12–23, Jan. 1995.

[14] A. M. Adas, "Using Adaptive Linear Prediction to Support Real-Time VBR Video Under RCBR Network Service Model," IEEE/ACM Trans. Net., vol. 6, no. 5, Oct. 1998, pp. 635–45.

[15] N. G. Duffield et al., "A Flexible Model for Resource Management in Virtual Private Networks," Proc. ACM SIGCOMM'99, Cambridge, MA, Oct. 1999, pp. 95–108.

[16] Z. Sahinoglu and S. Tekinay, "A Novel Approach Bandwidth Allocation: Wavelet-Decomposed Signal Energy Approach," Proc. IEEE GLOBECOM'01, San Antonio, TX, Dec. 2001.

[17] P. Siripongwutikorn, S. Banerjee, and D. Tipper. A Survey of Adaptive Bandwidth Control Algorithms, IEEE Communications Survey, vol. 5, no. 1, Third Quarter, 2003.

[18] J. Chandramohan, "An Analytic Multiservice Performance Model for a Digital Link with a Wide Class of Bandwidth Reservation Strategies," IEEE Journal on Selected Areas in Communications, Vol. 9, pp. 220-225, 1991.

[19] E. A. Feinberg and M. I. Rieman, \Optimality of Randomized Trunk Reservation," Probability in the Engineering and Informational Science, Vol. 8 pp.463-489, 1994.

[20] A. Girard and M. T. Ho, "Optimization of ISDN Networks with Partial Sharing," Proceedings of the 14th International Teletraffic Congress, Vol. 1a, pp. 1251-1260, 1994.

[21] S. Jordan and P. P. Varaiya, "Control of Multiple Service, Multiple Resource Communication Networks," IEEE Transactions on Communications, Vol. 42, pp. 2979-2988, 1994.

[22] J. S. Kaufman, "Blocking in a Shared Resource Environment," IEEE Transactions on Communications, COM-29, 1474-1481, 1981.

[23] Y. H. Kim and C. K. Un, "Analysis of Bandwidth Allocation Strategies with Access Restrictions in Broadband ISDN," IEEE Transactions on Communications, Vol. 4, pp. 1771-781, 1993.

[24] B. Kraimeche and M. Schwartz, "Analysis of Tra c Access Control Strategies in Integrated Service Networks," IEEE Transactions on Communications, COM-33, 1085-1093, 1985.

[25] B. Kraimeche and M. Schwartz, "Bandwidth Allocation Strategies in Wide-Band Integrated Networks," IEEE Journal on Selected Areas in Communications, SAC-4, pp. 869-878, 1986.

[26] [17] B. Kraimeche and M. Schwartz, "A Channel Access Structure for Wideband ISDN," IEEE Journal on Selected Areas in Communications, SAC-5, pp. 1327-1335, 1987.

[27] C.-T. Lea and A. Alyatama, "Bandwidth Quantization and States Reduction in the Broadband ISDN," IEEE/ACM Transactions on Networking, Vol. 3., No. 3., June, 1995.

[28] D. Medhi, A. van de Lievoort and C. S. Reece, "Performance Analysis of a Digital Link with Heterogeneous Multislot Tra c," IEEE Transactions on Communications, Vol. 43, pp. 968-976, 1995.

[29] T. Oda and Y. Watanabe, "Optimal Trunk Reservation for a Group with Multislot Traffic Streams," IEEE Transactions on Communications, Vol. 38, pp. 1078-1084, 1990.

[30] K. W. Ross, Multiservice Loss Models for Broadband Telecommunication Networks, Springer-Verlag, London, 1995.

[31] K. Yamnouchi and T. Oda, "Adaptive Circuit Access Control for Network Resource Management," IEICE Trans. on Comm., Vol. E78-B, No. 3, pp. 303-310, March 95.

[32] A. Arvidsson, "Management of Reconfigurable Virtual Path Networks," Proc. of 14th International Teletraffic Congress, pp. 931-940. 1994.

[33] J. Filipiak, and P. Chemouil, "Modeling and Prediction of Traffic Fluctuations in Telephone Networks," IEEE Trans. on Communications, Vol. Comm-35, No. 9, pp. 931-941, 1983.

[34] R. Burden, J.D. Faires, and A. C. Reynolds, Numerical Analysis, Second Edition. Prindle,Weber & Schmidt, Boston, MA, 1981.

[35] S. Eick, W. Massey, and W. Whitt, "Nonstationarity in Offered Traffic to the AT & T Long Distance Network," AT & T Symposium on Performance Analysis and Teltraffic Restoration: Theory and Application. Homedel, N.J., June 1990.

[36] G. Gopal, C. Kim, and A. Weinrib, "Algorithms for Reconfigurable Networks," Proceedings of 13th International Teletrafficc Congress, pp. 341-347, 1991.

[37] B. Groskinsky, D. Medhi and D. Tipper,"An Investigation Of Adaptive Capacity Control Schemes In A Dynamic Traffic Environment", IEICE Tran. on Communications, Vol. E84-B, pp. 263-274, February 2001.

[38] X. Gu, K. Sohraby, and D. Vaman, Control and Performance in Packet, Circuit, and ATM Networks, Kluwer Academic Publishers, Norwell, MA, 1995.

[39] D. Jagerman, "Nonstationary Blocking in Telephone Traffic," Bell System Technical Journal, Vol. 54, pp.625-661, 1975.

[40] M. Vidyasagar, Nonlinear Systems Analysis, Printice-Hall, Inc, Englewood Cliffs, NJ, 1978.

[41] M. Logothetis, and S. Shioda, "Medium-Term Centralized Virtual-Path Bandwidth Control Based on Traffic Measurements," IEEE Transactions on Communications, Vol. 43, No. 10, 2630-2640, 1995.

[42] D. Medhi, "Multi-hour Network Design for Dynamically Reconfigurable Wide-Area ATM Networks," IEEE/ACM Trans. on Networking, Vol. 3, pp. 808-819, 1995.

[43] V. Paxson and S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling," IEEE/ACM Trans. on Networking, Vol. 3, pp. 226-244, 1995.

[44] A. Pitsillides, P. Ioannou, and D. Tipper, (1996) "Integrated Control of Connection Admission, Flow Rate, and Bandwidth for ATM Based Networks," Proc. IEEE INFOCOM'96, pp. 6d3.1-6d3-9, 1996.

[45] R. Palazzo, "A Lyapunov Approach to Dynamic Bandwidth Allocation in B-ISDN", M.S. thesis, Clemson University, 1994.

[46] Y. Qian, D.Tipper, and D. Medhi, "An Analysis of Access Control Schemes for Multirate Loss Networks Under Nonstationary Conditions," Proc. IEEE INFOCOM'96, pp 730-737, 1996.

[47] S. Shioda, H. Toyoizumi, H. Yokoi, T. Tsuchiya, and H. Saito, "Self-sizing Network: A New Concept Based on Autonomous Network VP Bandwidth Adjustment," Proceedings of 15th International Teletraffic Congress, pp. 997-1006, 1997.

[48] D. Tipper, M.K. Sundareshan, M.K. "Numerical Methods for Modeling Computer Networks Under Nonstationary Conditions," IEEE Journal on Selected Areas in Communications, Vol. 8, No. 9, pp.1682-1695, 1990.

[49] W. Wang, D. Tipper, and S. Banerjee, "A Simple Approximation for Modeling Nonstationary Queues," Proc. IEEE INFOCOM'96, pp. 2d.3.1-2d.3.8, 1996.